Virtual Try-On Creation using Generative-AI

BY

TEO ZI NING

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Virtual Try-On Using Generative-AI

Academic Session: _JUNE 2024__

I ______________________TEO ZI NING____________________________

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)	(Supervisor's signature)
Address:	
No 8, Jalan Tembusu 13,______	
Taman Rinting,_____________	Tan Hung Khoon
81750 Masai, Johor__________	Supervisor's name
Date: ___11 September 2024_____	**Date**: 11/9/2024

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 11 September 2024

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***TEO ZI NING*** (ID No: 2104495) has completed this final year project/ dissertation/ thesis* entitled "*Virtual Try On Creation using Generative-AI* " under the supervision of Ts Dr Tan Hung Khoon (Supervisor) from the Department of Computer Science, Faculty of Information and Technology , and Dr Tan Joi San (Co-Supervisor) from the Department of Computer Science, Faculty of Information and Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

(*TEO ZI NING*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**VRITUAL TRY-ON USING GENERATIVE-AI**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : TEO ZI NING

Date : 11 September 2024

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Ts Dr Tan Hung Khoon who has given me this bright opportunity to engage in this project. It is my first step to establish a virtual try-on. A million thanks to you.

To a very special person in my life, Chin Chui Ling, for her patience, unconditional support, and love, and for standing by my side during hard times. I would also like to thanks to my project collaborator, Chen Sin Yee, where helping each other to develop the application together in these days. Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

In today's rapidly changing world, technology and fashion have come together in exciting new ways. One of the most innovative developments is virtual try-on applications. These apps let users try on clothes digitally without needing to go to physical stores. This has become especially popular during the COVID-19 pandemic, as it provides a safe and easy way for people to shop for clothes from the comfort of their homes. Instead of going to a store to see how an outfit looks, users can simply upload a picture of themselves or use their camera to see what the clothes would look like on them in real time.One of the key reasons why these apps are becoming so popular is the use of advanced technology, particularly generative artificial intelligence (AI). For example, some virtual try-on applications use image-based virtual try-ons that are powered by generative adversarial networks (GANs). These networks help create realistic images that show how clothes would look on a person. In addition, augmented reality (AR) and virtual reality (VR) are being used to make the virtual try-on experience even more interactive. This project will focus on developing a seamless and intuitive virtual try-on experience that appeals to modern consumers. With the power of AI and virtual reality, users will be able to try on clothes, experiment with new styles, and make purchasing decisions in a way that's convenient, fun, and perfectly suited for the digital age.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligent
GAN	Generative Adversarial Networks
AR	Augmented Reality
VR	Virtual Reality
CAGAN	Conditional Analogy Generative Adversarial Network
VITON	Virtual Try-On Network
SSIM	Structural Similarity Index Measure
LPIPS	Learned Perceptual Image Patch Similarity
FID	Fréchet Inception Distance

Chapter 1 Introduction

The way that customers engage with different kind of apparel and accessories has undergone a revolutionary change in recent years due to the convergence of fashion and technology especially throughout the pandemic. Approximately, 70% of consumers think that buying suitable clothes online is hard as it could not be tried, and it leads to return a $550 billion problem for business [1]. In order to bridge the gap between traditional brick-and-mortar shopping and the digital shop, virtual try on, a technology-driven solution that applying unique generative-Artificial Intelligence (AI) has emerged as a potential option where enable people to make use of the capability to digitally "try on" garments and view their looks. For instance, H&M, a clothes branding has unveiled the idea of a virtual dressing room which allow shoppers to utilize the augmented reality technology to try on virtual apparel which highly increase the user experience [2].

The underlying concepts of virtual try on are based on the idea of clothing transfer, which is seamless adoption of clothes from one picture onto a user's image while maintaining both the garments are perfectly match to the user's body shape and position. It basically can be categorized into human pose estimation, garments parsing, garments landmarks detection and lastly fashion synthesis in order to achieve the product result produce by virtual try on system. The popularity of this idea has grown as a result of developments in generative AI especially in unsupervised learning method - Generative Adversarial Network (GAN), which has proven to be very adept at produce images that are both extremely realistic and visually compelling in several studies [3].

This project will develop a mobile application which called "Batik Creator" that utilizes the VITON-HD GAN model to showcase how virtual try-on systems can transform digital perceptions of clothing using advanced AI technology. Besides, the application also integrates features for batik transfer onto cloth, allowing users to apply traditional patterns digitally to their outfits, enabling people to interact with garments more convincingly online. Users can effortlessly experiment with various clothing styles and designs, gaining a more personalized shopping experience without leaving their homes. This combination of technology makes fashion exploration both fun and practical, enhancing the convenience of digital fashion.

1.1 Problem Statement and Motivation

The fashion industry is currently challenged by the limitations of traditional e-commerce platforms in providing a realistic shopping experience. Customers are unable to visualize how clothing will fit or appear on their bodies when shopping online, leading to decreased engagement, high return rates, and lower sales conversions [4]. Additionally, the demand for personalized and interactive shopping experiences has surged in the digital age, making it difficult for fashion brands to differentiate themselves in a saturated market. Despite the potential of advanced technologies like Generative Adversarial Networks (GANs), their adoption for virtual try-on technology remains underexplored, especially in complex clothing patterns such as batik. Addressing these challenges could bridge the gap between the physical and digital shopping experiences, providing customers with a more immersive and satisfying interaction with apparel. The motivation behind this project stems from the transformative potential of GAN-based virtual try-on technology, particularly tools like VITON-HD, in revolutionizing online apparel shopping. By allowing users to see lifelike representations of garments on their own bodies, virtual try-on systems promise to boost customer engagement, satisfaction, and sales for fashion brands. The integration of traditional clothing patterns like batik into this digital framework adds cultural richness and authenticity to the virtual shopping experience. This project aims to empower fashion businesses with innovative tools that leverage the power of GANs to provide personalized, dynamic, and culturally informed shopping experiences, ultimately helping them stand out in an increasingly competitive online marketplace.

1.2 Objectives

In this project, it is going to create an application that offers users a wide range of randomly generated batik patterns. By leveraging generative algorithms, users can explore a variety of unique and traditional batik designs which can be generated randomly. This feature will allow them to visualize different options and engage with the cultural richness of batik patterns. Additionally, the application also include batik cloth transfer function as a supporting feature which enable users to apply the batik onto a cloth. This transfer function will provide users with a more interactive and customizable experience in viewing batik patterns on a cloth.

Besides, another key objective of the project is the developed application aim to integrate a feature that allows users to visualize how the normal wearing cloth or batik-patterned cloth would look when worn onto a model. By overlaying the selected cloth onto images of models, the application will provide users with a realistic preview of how the clothing fits and appears with the chosen patterns. This feature will enhance the user's ability to make informed decisions about their clothing choices and ensure that the virtual try-on experience feels as authentic and interactive as possible.

Not only that, the project will implement the VITON-HD model, a state-of-the-art generative adversarial network (GAN) tailored for virtual try-on applications. By incorporating this advanced model, the application will simulate how clothing fits on a user's body with remarkable realism. The VITON-HD model will enable users to experience a seamless and lifelike try-on process that mirrors the real-world experience of trying on garments in a store. This virtual try-on feature will not only improve user engagement but also provide fashion businesses with a powerful tool to increase customer satisfaction and sales.

1.3 Project Scope and Direction

The scope of this project involves developing an innovative virtual try-on application that integrates several cutting-edge features to enhance the user experience in exploring batik patterns and clothing. The first key scope area focuses on the development of a Batik Pattern Generator. This feature will allow users to randomly generate a wide variety of traditional and modern batik patterns, giving them the freedom to explore diverse options before choosing one for their garment. The application will also include a Batik Cloth Transfer Function, which serves as a supporting feature, enabling users to visualize how selected batik patterns would appear on different types of clothing. This function ensures that users can engage more deeply with the rich cultural heritage of batik while personalizing their clothing choices.

The second scope is aimed on enhancing the User Visualization Experience. By leveraging advanced overlay techniques, the application will allow users to see how their chosen batik patterns would look on actual garments when worn by models. This interactive visualization tool will help users make more informed decisions, offering them a lifelike preview of how the clothing fits and appears, thus creating an engaging and immersive experience.

Finally, the project will implement the VITON-HD model as its core technology for the virtual try-on experience. This GAN model will provide users with a realistic simulation of how garments fit their bodies. By integrating VITON-HD, the application aims to offer a seamless virtual try-on process, making the user experience more dynamic, lifelike, and engaging.

Overall, the project direction is geared towards utilizing cutting-edge technology to blend cultural heritage with modern digital experiences. By combining the power of GAN-based virtual try-ons with interactive batik pattern visualization, this project aims to fill the gap in the fashion industry by offering a unique and personalized online shopping solution that resonates with today's consumers.

1.4 Contributions

This study explores the application of Generative Adversarial Networks (GANs) to advance virtual try-on systems within the fashion and apparel industry. By leveraging the unique capabilities of GAN technology, the project contributes to multiple areas of virtual try-on development. One key contribution is enhancing user interaction and engagement with virtual clothing models by refining the accuracy and realism of virtual clothing simulations. This includes addressing challenges in batik transfer, where garments are realistically rendered onto digital models, providing a more immersive and personalized virtual shopping experience.

An important aspect of this study involves pre-processing images before they are inputted into the GAN model which are resizing image into 768X1024 resolution and produce cloth mask. This pre-processing step is crucial for generate the try on result image as it ensures that the images meet the input requirements of the GAN. By standardizing and preparing images, the system ensures that the GAN can successfully generate the desired virtual try-on output without errors most of the time. This step plays a critical role in the overall effectiveness and efficiency of the virtual try-on process, enhancing the quality of the final results and ensuring that users experience a seamless and realistic virtual try-on session. Through these contributions, the project aims to fill gaps in the current virtual try-on landscape by integrating GAN-based solutions that improve the visual representation of clothing, enhance user experience, and offer greater customization possibilities.

1.5 Report Organization

The project’s details are shown in the following chapters. In Chapter 2, it shows the literature review on the GAN model and some system reviews. Then, a system architecture, use case diagram and activity diagram are shown in Chapter 3, while Chapter 4 is in charge of displaying the system design which include block diagram and component specifications. Furthermore, Chapter 5 contains the system implementation part while Chapter 6 contains the system testing part. Lastly, conclusion and recommendation is include in Chapter 7 and next is the References part, and there is an appendix at the back that includes weekly reports, plagiarism checks, and checklists.

Chapter 2 Literature Review

2.1 Previous Works

2.1.1 GAN Architecture

GAN is one of the generative machine learning models responsible for producing synthetic images that closely resemble real image. This was accomplished through a process of trial and error and was made by a discriminator whose job it was to evaluate the generated images' trustworthiness, ensuring they closely mimic real ones. GANs consist of two neural networks: the generator, tasked with creating fake images, and the discriminator which seek to distinguish real from fake images. In GANs, an adversarial loss function, often referred to as GAN loss, came into play, where the generator often tried to minimize the loss function. Inversely, generator attempted to maximize the loss function. The interaction between the generator and discriminator was illustrated in the figure below, where one worked to produce fake image while the other endeavoured to classify fake image until the image was able to be classified as a real image [5]:

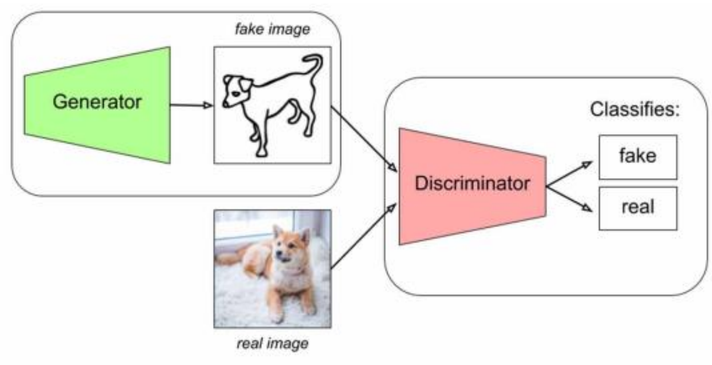


Figure 2.1.1 Generator and Discriminator in GAN

2.1.2 Early GAN-Based Approaches

Conditional Analogy Generative Adversarial Network (CAGAN) which was the first GAN model that being invented involved in virtual try-on. [6] suggested the idea of trading garments for the desired clothing by given certain human pairings. It had an encoder-decoder generator and a discriminator that provided classifications throughout the whole spatial field while the generator in charge of produced 4 channel output. Additionally, the definition of cycle loss was mentioned in the study where the concept briefed it would assist to strengthen the stability of the generator by guaranteeing that the other components of the image stayed the same while the target human's clothing were being changed. As the experiment result, it shows that the model able to produce photos that retained the appearance of human photographs and only exchanged the essential garment elements. However, it was unable to transfer textures and could only handle basic colored clothing.

2.1.3 Coarse-to-Fine Attire Transfer and Person Representation

The virtual try-on network (VITON), a coarse-to-fine architecture that effortlessly translated target clothing in a photo to the comparable region of a clothed person in an image, was the novel framework the author presented based on [7]. The architecture of VITON was similar to [8] which a multi-task encoder-decoder network was implemented to create a coarse synthesis dressed image wore the desired attire in the same position. The different characteristics of the model was that it comprised of a refinement network which was in charge of transferring the desired object with realistic deformations and intricate visual patterns with primary purpose of composited the deformed attire item to the coarse image. To achieve the objective, a clothing-agnostic person representation was introduced, encompassed a range of features such as body parts, face, hat and pose which served as a prior constraint for the synthesis process where the encoder-decoder would be trained to transfer the target attire while considering the person’s representation. The outcomes demonstrated that VITON was able to produce image that maintained the target human skin tone. Additionally, it was able to transfer short-sleeved shirt in which the arms were correctly synthesized despite the fact that they were no visible in the original image.

2.1.4 Enhanced Attire Transfer with Optimization

[7] proposed an algorithm, TryOnGAN, that could outperform SOTA by focusing on optimizing the loss function for automated computing the interpolation coefficient for each layer to ensure the target person identity such as hair, skin and body, were preserved while only the garment was rendered. It was supported to StyleGAN-based structure where the model was trained with unpair data compared to [7]-[9]. The algorithm proposed in the study was divided into two phases, the first of which involved training a pose-conditioned StyleGAN2 - also referred to as a generator to create a photorealistic image of a person wearing a specific output by taking two input images and the pose heatmap encoder as input. In order to properly project the images into the latent space of generator, an optimizer used a pair of input images to produce latent codes that minimized the perceptual gap between the input image and the generated image (referred to as two input images). The second stage was in charge of layer-by-layer interpolation coefficient optimization. Additionally, TryOnGAN incorporated two discriminators: a pose discriminator and a segmentation discriminator, which received a pose heatmap and segmentation image as input, respectively. Once the model had been trained, comparisons with other GAN models such as ADGAN, CP-VTON and ACGPN were made in order to evaluate TryOnGAN in both quantitative and qualitative ways. The results showed that TryOnGAN performed better in producing the arms area and preserved body shape and skin color in the image in photorealism.

2.1.5 Single Image Attire Transfer with Textual Descriptions

[8] proposed a framework that implemented two main stages to produce the virtual try-on results image where the first stage represented of generated segmentation map based on person’s body shape and code which was a sentence description of a new desired attire in order to ensure the image was generated perfectly aligned with the target person’s pose and body structure, while second stage performed rendered the attire to the person based on generated segmentation map and design code. The whole process of second stage was known as compositional mapping where the attire was created with its own texture and then the texture was combined later to generate the final image represent of the person wore on the designed attire. Hence, it was shown that the framework trained two generators where one responsible for segmentation map generation while the left one in charge of texture rending while the discriminator was supplied of same conditions which were segmentation maps and designed

code where to ensure the consistency between the requirement and the discriminative networks. The framework was then evaluated by using a comparison technique with several baselines, including one-step GAN which generated images directly in a single step within GAN framework, non-compositional whose function was to directly generate the entire image as whole in a single stage without the need for segmentation guidance, and also 2D non-parametric. In the outcomes, the framework displayed less artifacts, greater visual quality, and clearer attire zones than these baselines, according to the comparative results between them.

2.1.6 Style-Based Flow for Garment Wrapping

[9] introduced a unique style-based appearance flow approach for virtual try-on garment warping. It improved robustness against significant misalignments between human and garment photos, especially in 'in-the-wild' settings with genuine postures. The proposed architecture method was started by estimating a coarse appearance flow by style modulation, then refined it through local feature correspondence. Besides, the method was being made to be parse-free model in order to minimize the detrimental effect of incorrect human parsing where the approach was focused on the layout of the component that wrapped the garments. The findings reveal that this model regularly generates high-quality and aligned try-on photographs, demonstrating its robustness to misalignments and tough circumstances. The model consistently generated excellent try-on images, including in difficult cases with hard postures and occlusions.

2.1.7 High-Resolution and Misalignment-Aware Normalization

The goal of the image-based virtual try-on is to accurately place apparel products on people in digital photos—a process that is essential to ensuring that online shoppers are satisfied. Unfortunately, the low-resolution results (e.g., 256×192) of the current approaches are hampered by issues such as visible artefacts in misaligned regions and problems preserving texture sharpness. To address this, this work presents VITON-HD, a revolutionary virtual try-on technique that can create high-resolution (1024 x 768 pixels) pictures. By utilizing a human representation that is not dependent on the genuine apparel worn by the subject, VITON-HD enhances adaptability, and in order to preserve garment details in high-resolution outputs, ALIAS normalizations and ALIAS generators were introduced to fix the misalignment problems. Comparative tests show that VITON-HD performs better than other options in terms of image quality, which represents a major breakthrough for virtual try-ona technology. Among the highlights of this study are the introduction of VITON-HD, which is the first model to successfully synthesize 1024 x 768 pictures, and the creation of novel approaches to address issues with apparel reliance and misalignment in virtual try-on systems [10].

2.1.8 Strengths and Weakness

In current project, VITON-HD was being utilized and based on the research of the VITON-HD, one of the VITON-HD model's strength is that it can generate high-resolution (1024 x 768) pictures, which is a notable improvement above current low-resolution output among to others GAN model. This improvement improves the user experience overall by enabling more realistic and detailed virtual clothes trials. Besides, the adoption of a garment-agnostic human representation also represents one of the main characteristics of VITON-HD. It helps in reduce reliance on the original clothing worn by persons and promotes adaptability. With this method, virtual try-on technology may be used to more situations and fashion products, allowing for more applicability across different types and designs of apparel. Additionally, VITON-HD could successfully resolve frequent problems related to virtual try-on, including misalignment problems between the intended clothing areas and the deformed apparel. There are also the use of ALIAS generator and normalization procedures greatly enhances the overall image quality and preservation of garment detail, resulting in more realistic and aesthetically pleasing virtual try-on outcomes. Nevertheless, there are potential drawbacks to the method's advantages. The synthesis of higher-resolution pictures might present challenges due to its computational complexity and resource needs, which could restrict accessibility and scalability in situations with limited resources. Furthermore, further testing and validation across a range of datasets and real-world situations are necessary to evaluate VITON-HD's robustness and generalizability in full, guaranteeing its applicability and dependability in a variety of fashion industry contexts [10].

2.1.9 Comparison Between VITON-HD and Other Models

	256 × 192			512 × 384			1024 × 768		
	SSIM↑	LPIPS↓	FID↓	SSIM↑	LPIPS↓	FID↓	SSIM↑	LPIPS↓	FID↓
CP-VTON	0.739	0.159	56.23	0.791	0.141	31.96	0.786	0.158	43.28
ACGPN	0.842	0.064	**26.45**	0.863	0.067	15.22	0.856	0.102	43.39
VITON-HD*	-	-	-	-	-	-	0.893	0.054	12.47
VITON-HD	**0.844**	**0.062**	27.83	**0.870**	**0.052**	**14.05**	**0.895**	**0.053**	**11.74**

Figure 2.1.2 Quantitative Comparison between VITON-HD and other models

From Figure 2.1.2, which was taken from [10], it is shown that the measurements of the VITON-HD model have a higher Structural Similarity Index Measure (SSIM) value, which is a statistic for estimating how similar two photos are to one another. Besides, VITON-HD also obtains lower Learned Perceptual Image Patch Similarity (LPIPS), which is a statistic used to compare two photographs' perceived resemblance, and the Fréchet Inception Distance (FID) value, which is a statistic used to gauge perception. Similarity between two pictures is measured by a metric called Tt, which is frequently used to assess the quality of generative models, especially in the context of GANs compared to others. The higher SSIM and lower LPIPS and FID imply a better sign. So, VITON-HD is being chosen as the GAN model being used in the project.

2.2 System Reviews

Application/System	**HeyBeauty**	**Style.Me**	**PICTOFit**
Platform Availability	Web, browser-based	Web Application	IOS, web-based
AI Integration	Yes, for size prediction and style	Yes, for size fit and recommendations	Yes, for 3D virtual try-on
User Interface	Intuitive but requires image upload	User-friendly, Mobile accessible	Intuitive, High
Key Strengths	Personalized style and size recommendations	Precise size fit, User insights	Realistic 3D avatar creation, AR support
Main Weakness	Limited by user input requirements	Limited to web application	Only available on IOS

Table 2.2.1 System Reviews of Different Applications/System

Table 2.2 shows the system reviews of HeyBeauty, Style.Me and PICTOFit where these three are virtual try on system currently at the market. HeyBeauty is an AI-driven platform designed to enhance the online shopping experience by allowing users to try on clothes virtually. The app creates a virtual dressing room where allow users to upload images or input their body measurements to generate a 3D model for trying on different outfits. It uses advanced AI algorithms to predict the best size for users across multiple brands and styles which helping to reduce returns and increase customer satisfaction. Besides, it also provides personalized style recommendations based on users' past shopping habits, making it a highly customized shopping experience. The app is accessible via web browsers and is geared toward both retailers and individual shoppers looking to improve their fashion experience online. However, HeyBeauty does have some limitations. While it offers a realistic and interactive experience, it requires users to upload images or provide detailed measurements, which may deter some from using the app. Nonetheless, HeyBeauty remains a powerful tool for those looking to enhance their online shopping experience with a high degree of customization and accuracy [11].

Style.Me is another leading virtual fitting room application that connects traditional in-store shopping experiences with online shopping through AI technology. It uses precise body mapping algorithms to help users find their perfect size fit and offers personalized style recommendations. The app is user-friendly and allows users to access it directly via their smartphones. Style.Me focuses on both male and female clothing and provides valuable advice on outfit matching, enhancing customer satisfaction and reducing return rates by ensuring the right fit from the start. It also collects analytical customer insights, making it a valuable tool for targeted marketing campaigns.One of the limitations of Style.Me is that it is currently only available as a web application, which might limit its accessibility compared to apps available on multiple platforms. Additionally, while it offers detailed size recommendations, it does not support offline use or integration with users' existing wardrobe items. Despite these drawbacks, Style.Me is a robust choice for those looking to improve their online shopping experience by trying on clothes virtually before purchasing [12].

PICTOFiT Shopping provides an advanced virtual try-on experience by utilizing AR (Augmented Reality) and AI technologies. It allows users to create photorealistic avatars and try on clothes virtually, offering a 3D view of how outfits will look and fit in real time. The app is highly interactive and supports retailers by integrating directly with web shops, helping to boost online sales and engagement by reducing return rates and enhancing the overall shopping experience. PICTOFiT is primarily available on iOS devices and can be accessed through its web platform.However, PICTOFiT is limited by its lack of compatibility with Android devices, which restricts its user base. Additionally, while it offers a visually engaging and realistic experience, it may require more processing power and a stable internet connection, which could be a barrier for some users. Nevertheless, PICTOFiT remains a top choice for retailers and customers seeking a seamless, engaging online shopping experience that closely mimics in-store fitting rooms [13].

Chapter 3 System Methodology/Approach

3.1 System Architecture Diagram

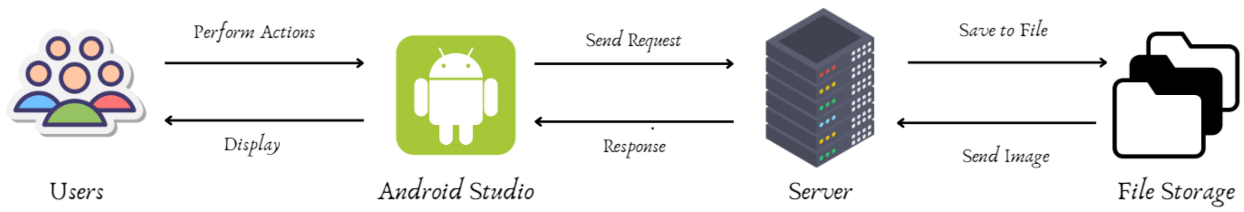


Figure 3.1.1 System Architecture Diagram of Batik Creator Application

The system architecture diagram of Batik Creator outlines how the system is divided into four primary components: **Users**, **Android Studio**, **Server**, and **File Storage**. Each component has a specific role and function that contributes to the overall operation of the application.

The **Users** represent the individuals who interact directly with the application through Android Studio. They can perform several actions within the app. These actions include generating random batik patterns from a variety of designs, trying different batik patterns on virtual clothing models, customizing their own batik designs by painting directly on images, and performing virtual try-ons to see how the batik designs would look on different garments. This interactive process allows users to personalize and visualize their designs before making any decisions or modifications.

Android Studio acts as the interface between the users and the back-end systems. Its primary responsibility is to display the user interface, allowing users to interact with the app intuitively and efficiently. Android Studio captures the users' inputs and converts them into action requests that are sent to the server. In return, it receives responses from the server and presents them to the user, such as displaying the generated virtual try-on result or batik patterns. Essentially, Android Studio bridges the communication between the user and the server, making sure that the app remains responsive and user-friendly.

On the **Server** side, the main function is to handle and process image data. When the user requests specific actions, such as generating a batik pattern or performing a virtual try-on, the server processes these requests. It analyzes the input data, processes the images, and generates the results. For example, if the user generates a virtual try on result, the server will perform pre-processing on the image and then just pass to the model for generate the result and sends the result image back to Android Studio. The server is also responsible for saving these processed images to the file storage system, ensuring that all generated images are stored for future use.

The **File Storage** component plays a vital role in saving and organizing the processed images. Once the server generates a result, such as a batik simulation or a virtual try-on result image, the image is saved to the file storage system. This allows the server to retrieve the stored images whenever needed, which for displaying them back to the user in Android Studio. The file storage ensures that users' designs and actions are preserved and can be accessed later, maintaining continuity in the application.

Together, these components form an efficient and streamlined system for users to create, customize, and visualize batik designs, with each part of the architecture contributing to a smooth and responsive user experience.

3.2 Use Case Diagram

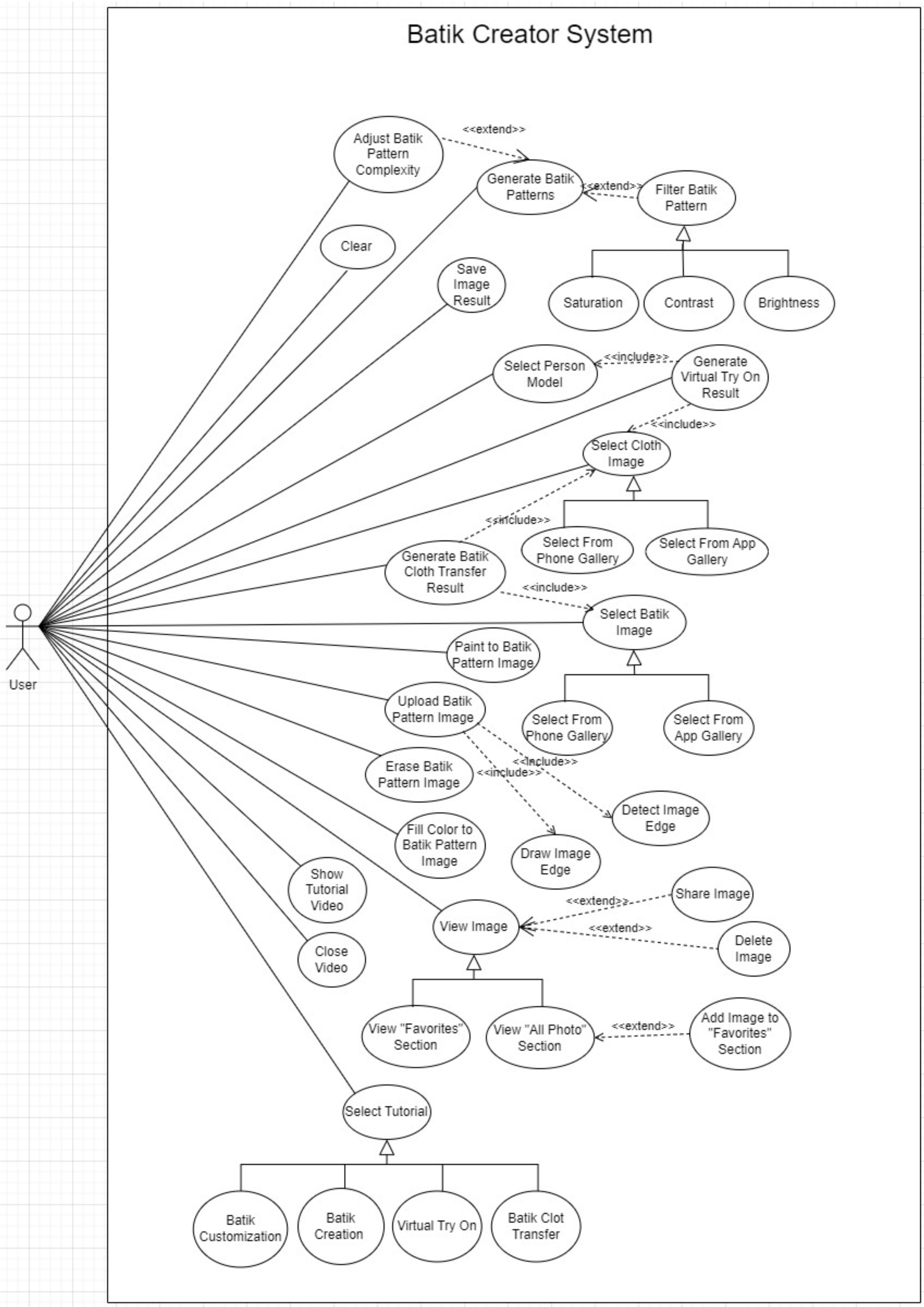


Figure 3.2.1 Use Case Diagram of Batik Creator

3.3 Activity Diagram

Use Case: Generate Virtual Try-On Result

This activity diagram shows the process on how to generate the virtual try-on result with initially getting user input which are cloth and person model. Then, the images will be send to the server for processing and a final result image will be passed back by the server to the user and display on application.

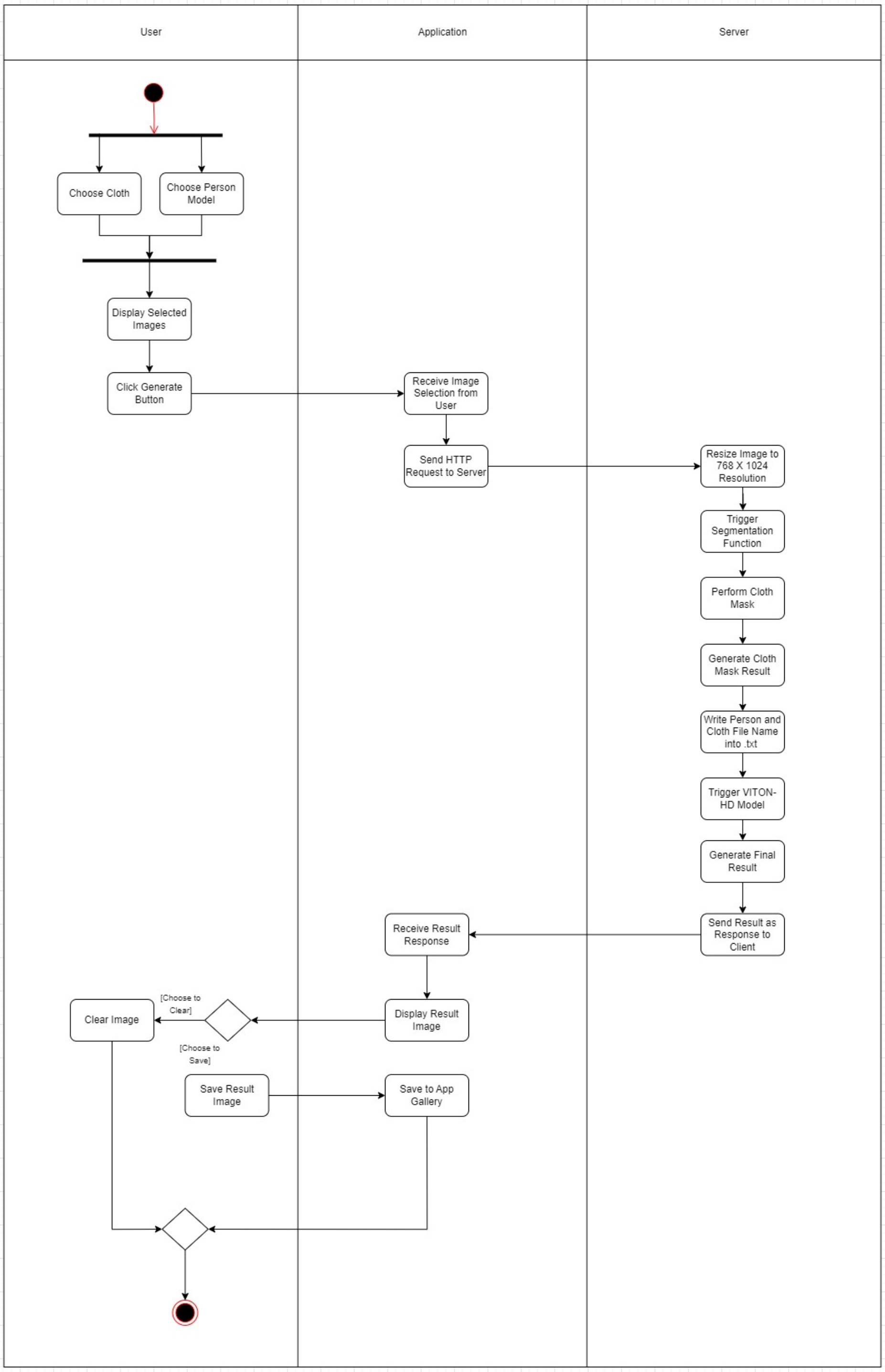


Figure 3.3.1 Activity Diagram of Virtual Try On

Use Case: Generate Batik Cloth Transfer Result

In order to have a batik on the cloth, the user has to choose the desired batik, whether from the app gallery or phone gallery, while also selecting a cloth image. After images are selected, the application will process the final result by applying some texture transform logic to the application.

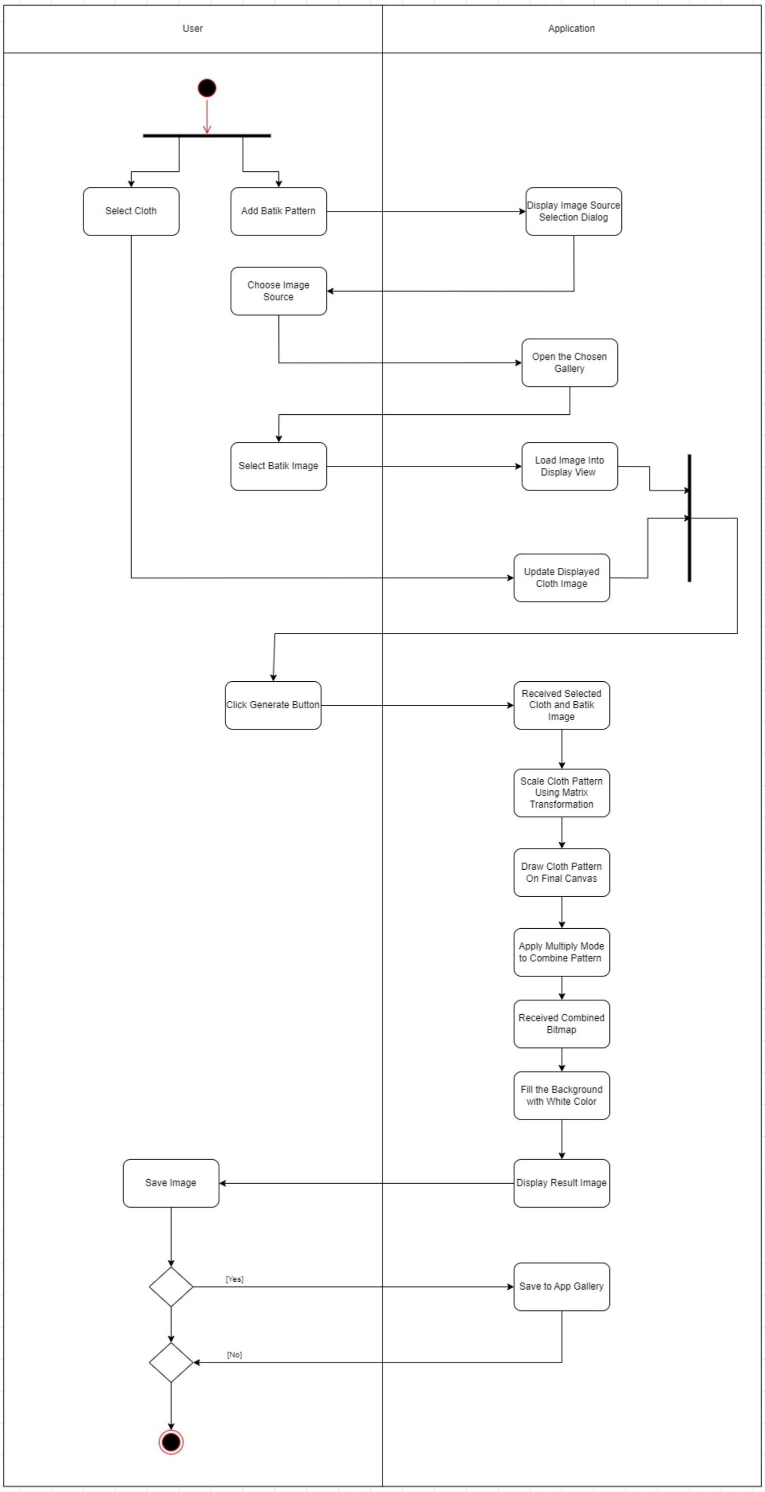


Figure 3.3.2 Activity Diagram of Batik Cloth Transfer

Use Case: Click Batik Patterns

The below diagram shows the process of how to generate a variety range of batik patterns randomly. Initially, the user has to click on the generate button to trigger the server to generate batik randomly and pass it back to the application. The received batik can be further filtered and adjusted for complexity.

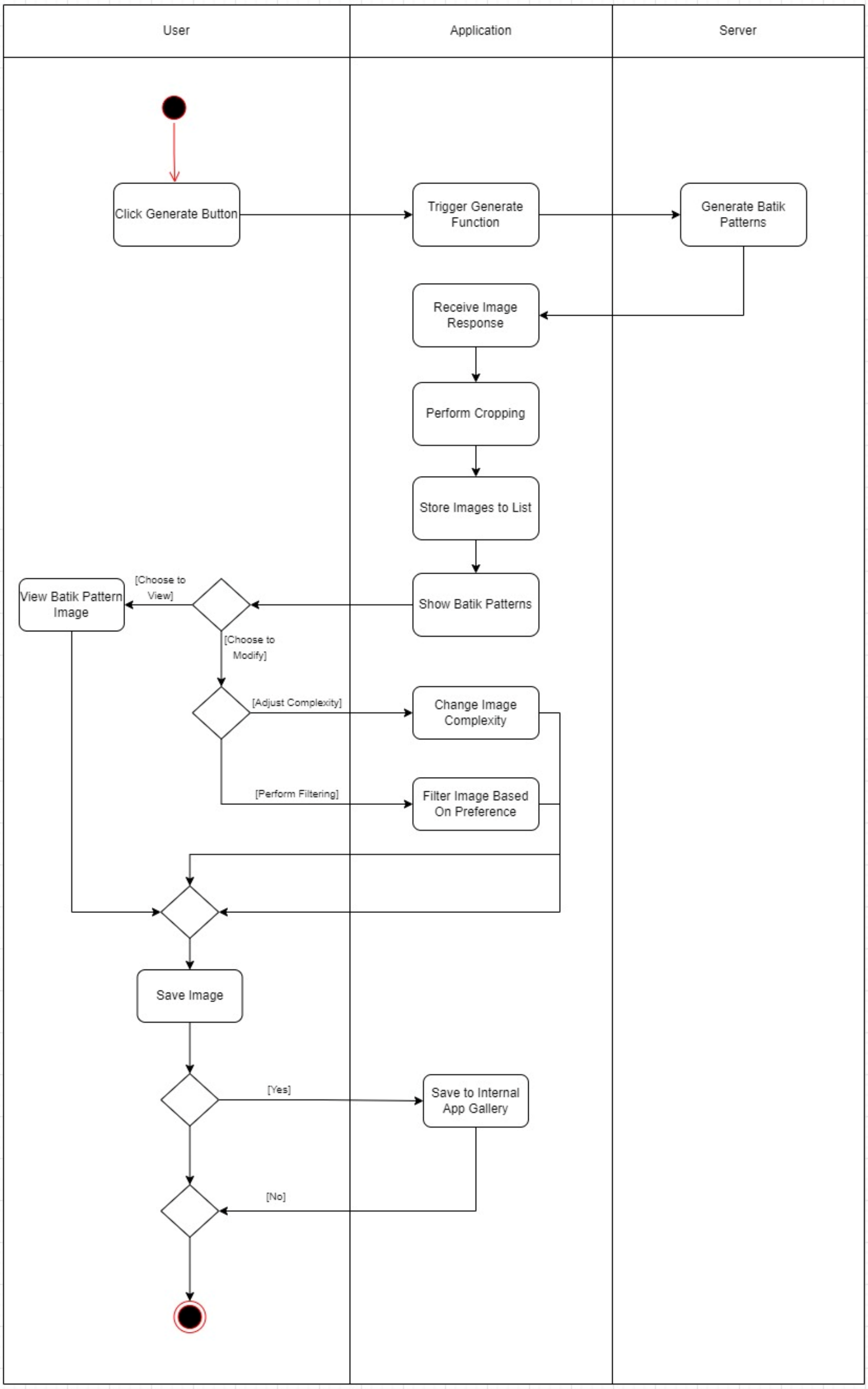


Figure 3.3.3 Activity Diagram of Generate Batik Patterns

Use Case: Customize Batik Patterns

Figure 3.3.4 shows the customization activity of batik patterns. The user has to select the image that they want to customize, and then the image will be processed by certain techniques in the application and final display on the screen.

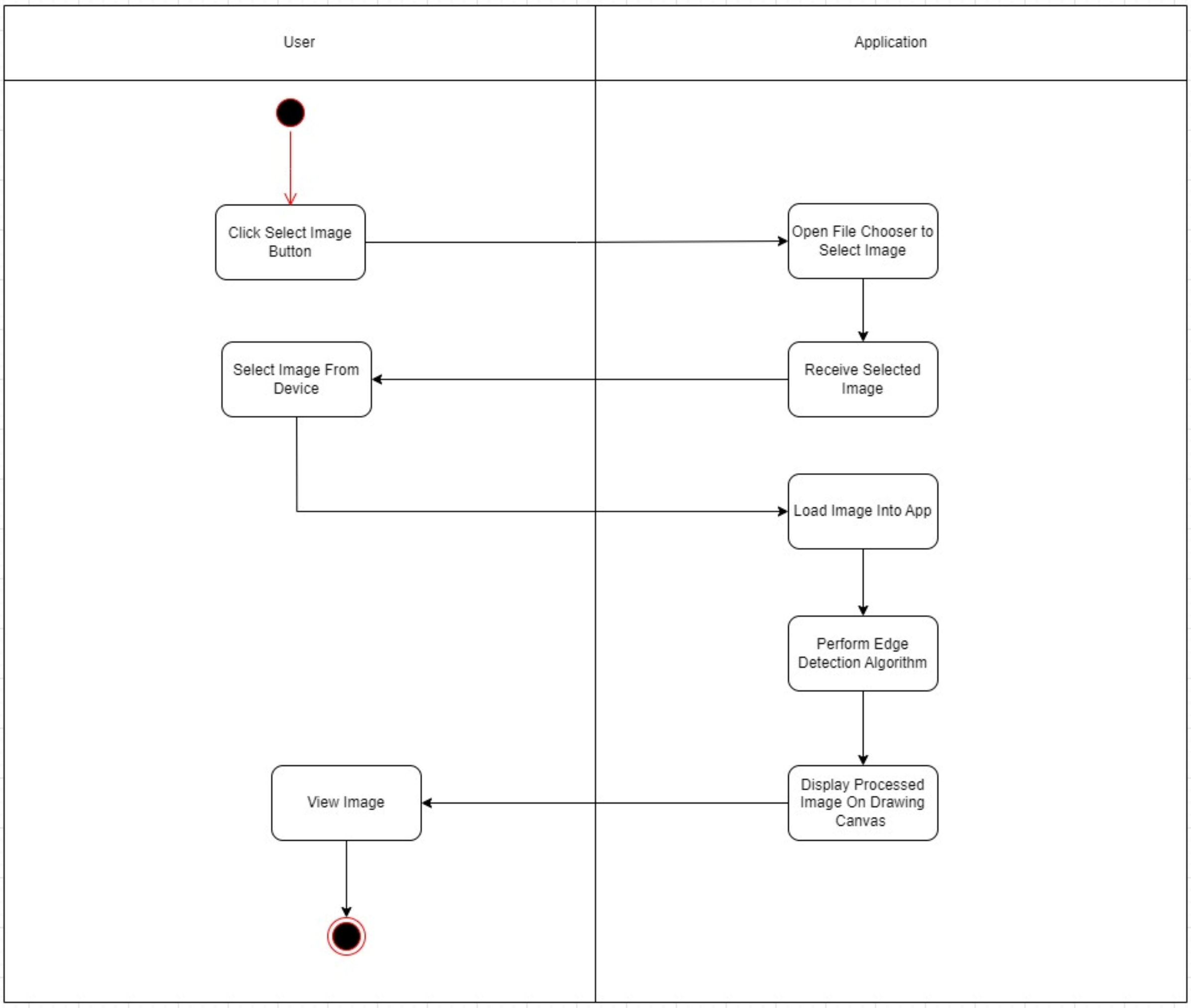


Figure 3.3.4 Activity Diagram of Customization of Batik Patterns

Use Case: View Image

Diagram below shows the flow of Gallery in the app. The user is able to view images that are stored in the internal gallery app with two tabs provided, which are “All Photos” and “Favorites.”. The user is able to add the image in the “All Photo” to “Favorites” if they like. Aside from this, galleries also have the function to let users share or delete images based on user preferences.

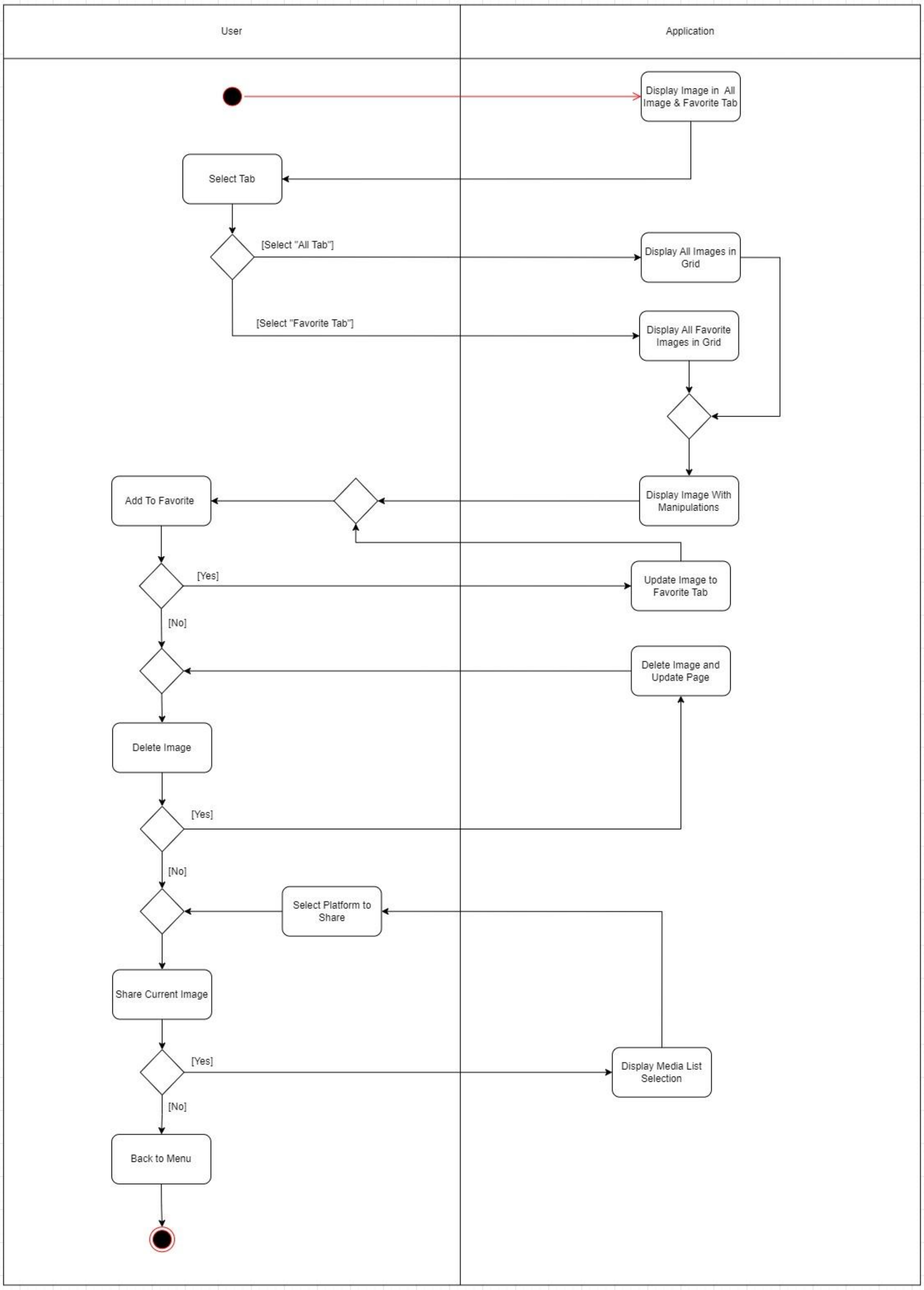


Figure 3.3.5 Activity Diagram of Gallery

Use Case: Select Tutorials

Figure below shows the tutorial flow of the Batik Creator. When the user clicks the tutorials button, it will prompt out a dialogue showing a few tutorial options where each of them represents a different function of the app. During the tutorial plays, the user able to pause the video while also closing the tutorial dialogue based on their preference.

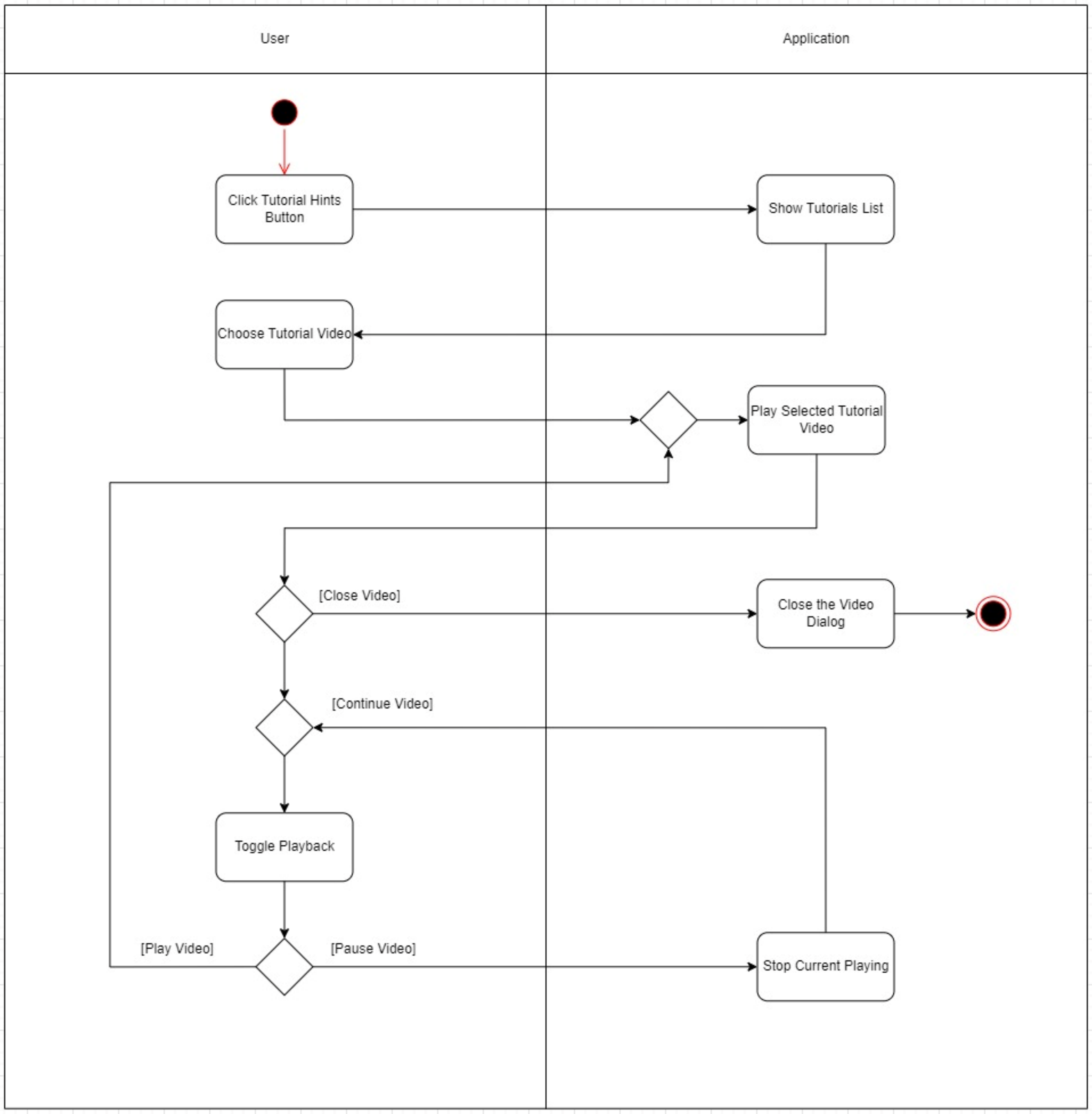


Figure 3.3.6 Activity Diagram of Tutorials

Use Case: Paint to Batik Pattern Image

Figure below shows the brush function in customization of batik. When users choose the brush icon in the app, they could brush the image, where at the same time the application is performing some technique to let the user be able to brush the image selected.

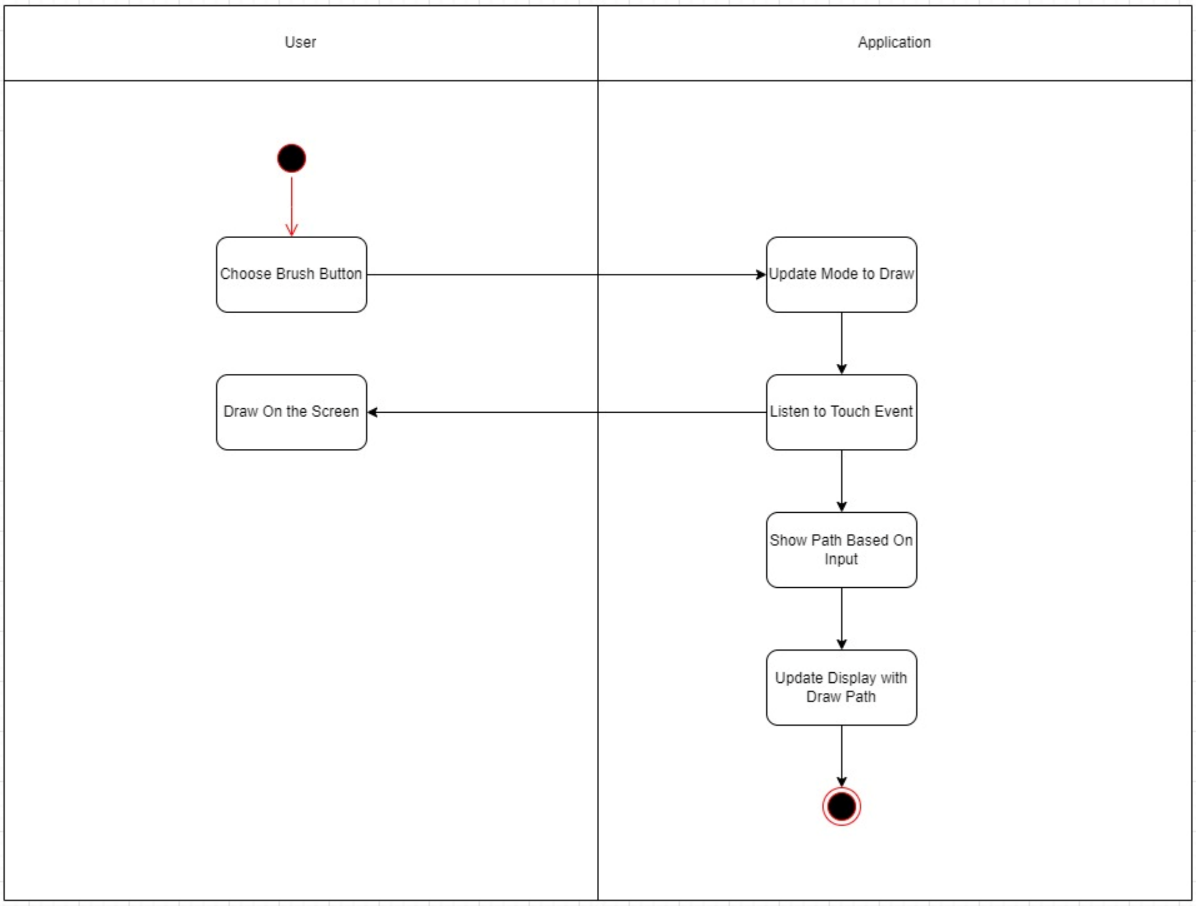


Figure 3.3.7 Paint Image

Use Case: Erase Batik Pattern Image

The below diagram shows the activity of erasing batik patterns in customization of batik pages. When users click on the erase icon, they could perform the erase function on the batik while also, at the same time, the application is listening to the touch event and processing a series of techniques to let users be able to erase the batik image.

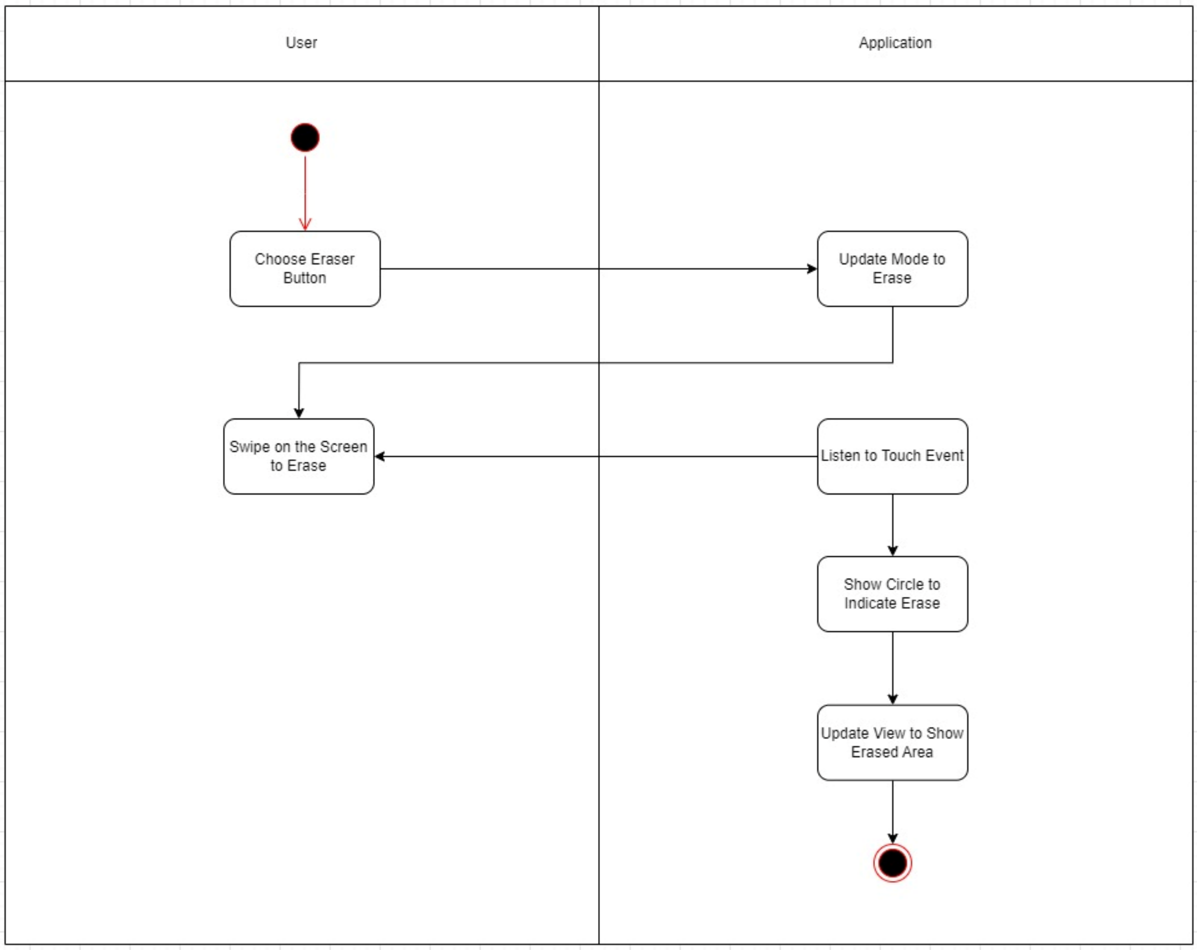


Figure 3.3.8 Activity Diagram of Customization Batik Patterns - Erase Image

Use Case: Fill Color to Batik Pattern Image

The diagram below shows the activity of filling in color to the batik pattern in the customization page. The user first has to select the fill color icon, and then the application will set into fill mode and carry out a series of techniques processed so that the user can fill in color to the batik afterwards.

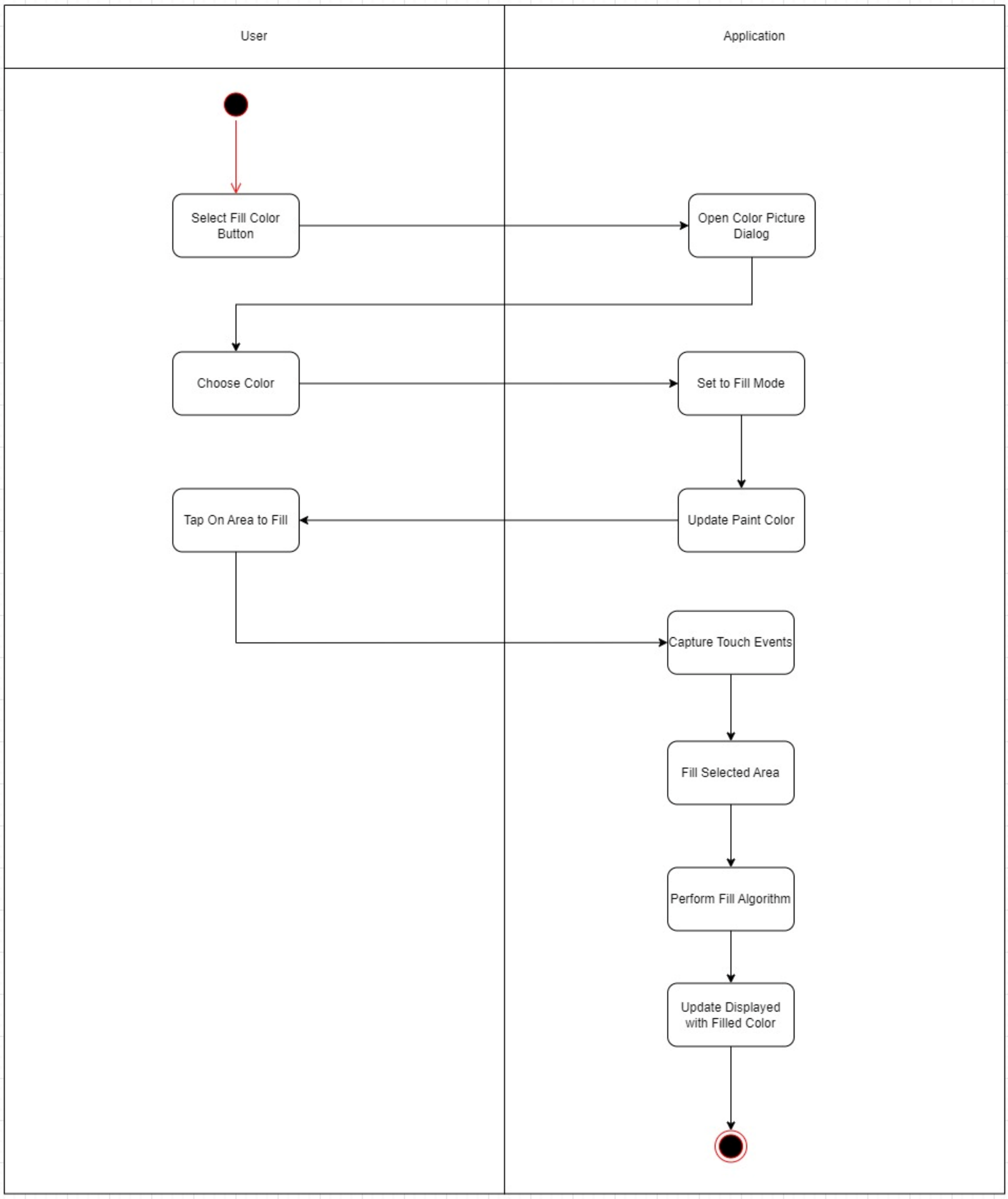


Figure 3.3.9 Activity Diagram of Customization Batik Patterns - Fill Image with Color

Chapter 4 System Design

4.1 System Block Diagram

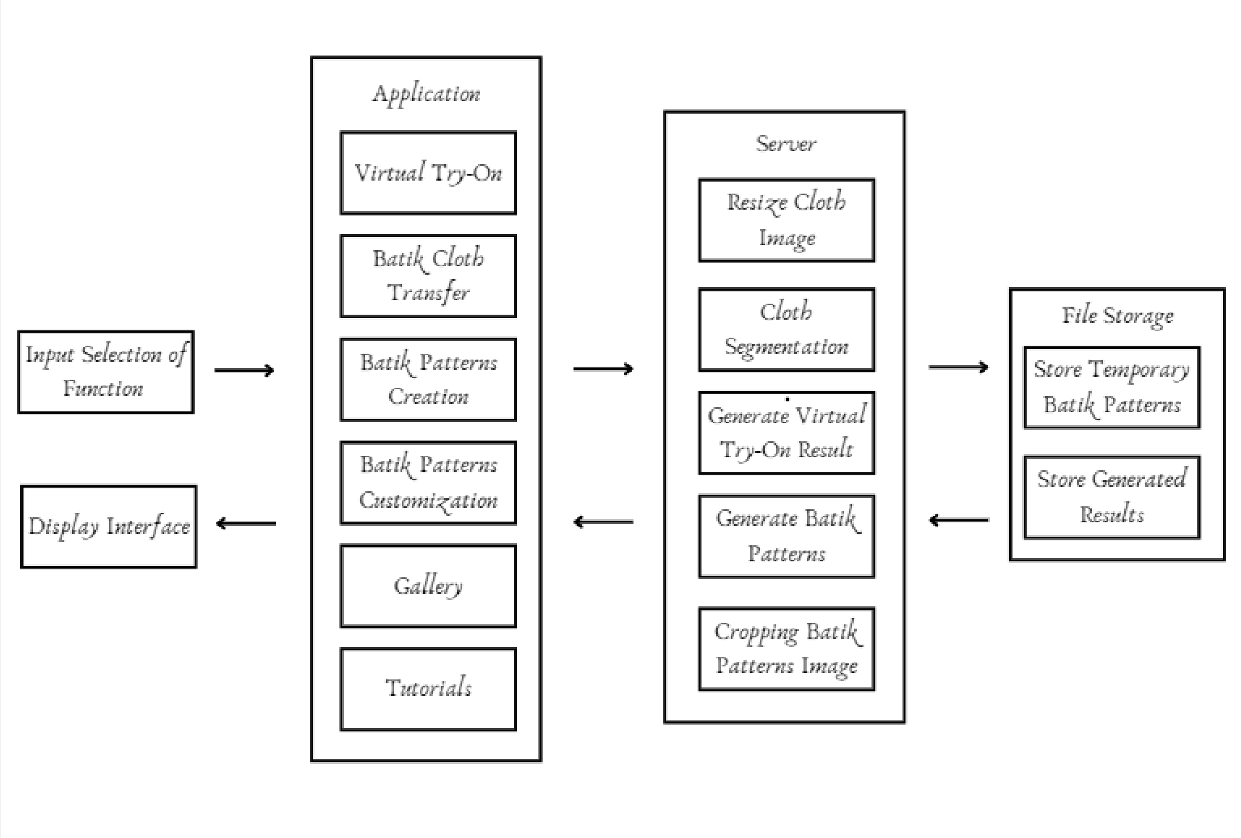


Figure 4.1.1 Block Diagram

Figure 4.1 shows the block diagram of the Batik Creator application. The main components of the application are performed virtual try-on, batik cloth transfers, batik pattern creation, batik pattern customization, galleries, and tutorials. These six components are the functions of the application. When users first start using the application, they need to select the function that they want to perform through the Android Studio. Android Studio acts as an intermediate platform that connects between backend server and users that contains a list of main functions that provide users with the ability to choose what function they want to perform and display the results to the user lastly. After the user chooses the intended function they want to perform, Android Studio will send a request to the server for processing the result. On the server side, it is in charge of doing generate batik patterns or virtual try-on results, resize images, cloth segmentation, and cropping batik patterns images, where each of the functions will be triggered based on what users want to perform at the Android Studio. For the file storage, it acts as a temporary storage, storing the batik patterns result, which needs to be further cropped and sent back to the Android Studio, while also storing the virtual try-on result image temporarily.

4.2 System Components Specifications

Virtual Try-On

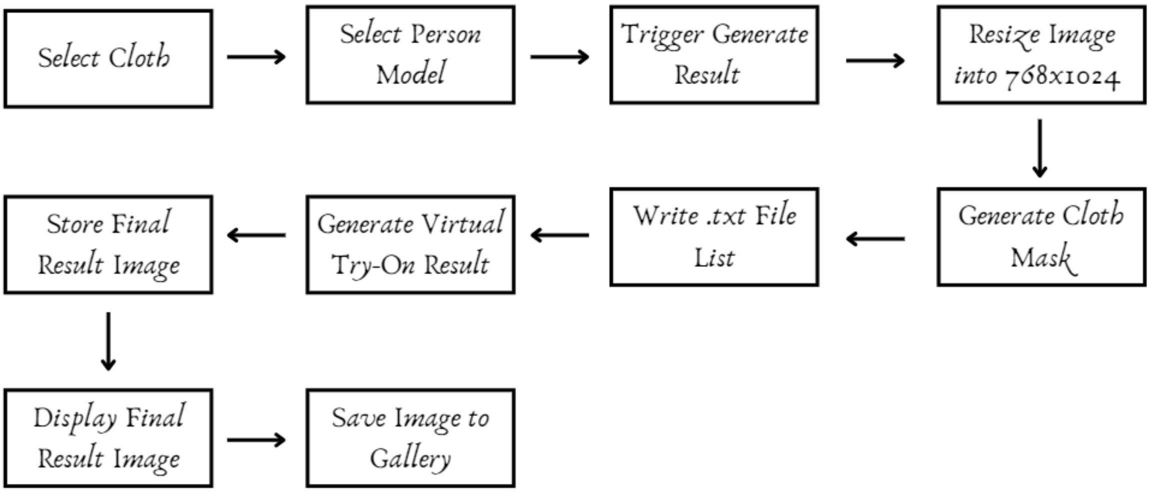


Figure 4.2.1 Component Specifications Diagram of Virtual Try-On

Figure 4.2.1 shows the details flow of the virtual try-on component. For the virtual try-on, users have to select a cloth and a person model in order to have a view of how a cloth looks on the person. The user may select the cloth from the internal app gallery or phone gallery, while for the person model, there are several options of person model that can be selected in the application that provide the user different views and favors. After selecting appropriate cloth and person model, the user may click the generate button, where the generate button will then trigger the server side to start processing the images selected by the users in order to get the result of the final outcome of selected cloth wear on selected person model. On the server side, it will start performing preprocessing on the images before calling the VITON-HD model for generating the result. The cloth image has to be first resized into 768x1024 resolution, as the GAN model will generate a better result if the image is in the size of 768x1024. After resizing the cloth image, it has also needed to perform cloth segmentation in order to get the cloth mask of cloth, where this is also the essential input source of the GAN model to generate the virtual try-on image. After both pre-processing methods are done, the server will just start writing a matching list of the cloth image filename and also the person filename into.txt, where this list is used for VITON-HD to perform matching the cloth onto the person model. Then, the VITON-HD will just start to generate the final result and store the final result in the file storage for temporary only. The application side will automatically retrieve the image stored in the file storage on the server through Flask and display the final result to the users at the interface.

Batik Cloth Transfer

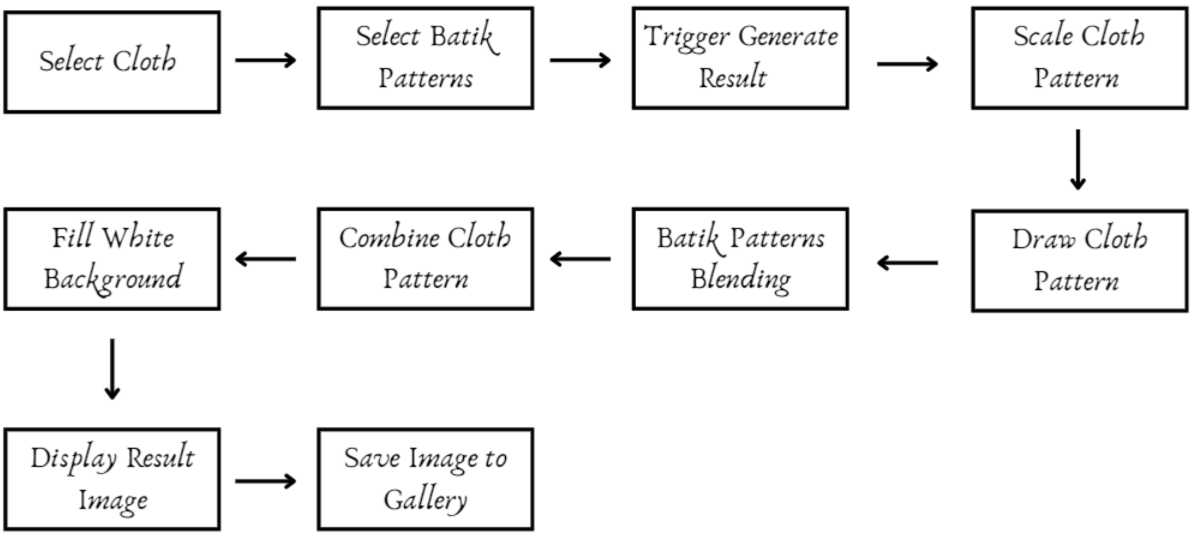


Figure 4.2.2 Component Specifications Diagram of Batik Cloth Transfer

Above figure shows the flow of the Batik Cloth Transfer component, where this component aims to provide a view to the user on what batik would look like on a cloth. When users intend to do batik cloth transfer, they need to select a batik pattern, which they may choose from the internal app gallery or phone gallery, while also needed to choose a cloth from the given variety of choices at the application. After choosing the wanted cloth and batik patterns, user may click the generate button to see the result of the batik applied to a cloth. When the generate button is triggered, the underlying side of the application will run the process of transforming batik onto cloth. The cloth and batik pattern will be first scaled by applying matrix transformation in order to make sure the batik pattern can be painted onto the cloth. Then, the cloth will be started to draw on canvas, and multiply mode will be applied for batik pattern blending. After that, the cloth and batik pattern are combined and applied with a white background and final result will be displayed to the user. The final generated result can be saved, where when the user clicks to save, the image will be saved to both the internal app gallery and the phone gallery. The image could be later used for a virtual try-on if the user wants to see how the cloth with batik looks on a person.

Batik Patterns Creation

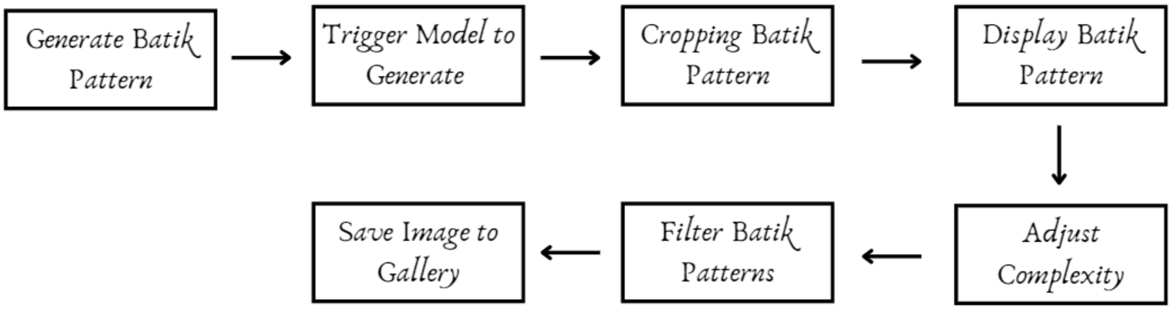


Figure 4.2.3 Component Specifications Diagram of Batik Patterns Creation

The above figure shows the overall flow of the batik pattern creation component. The user is able to click the generate button to generate the batik randomly initially. Then, the application side will trigger the server side to generate batik patterns, and the server side will start to generate batiks randomly and perform cropping on the batik patterns. After that, the result will be sent back and displayed on the application interface. The user may adjust the complexity and filter on the batik patterns following. Lastly, the user may save the batik pattern image to both the internal app and phone gallery if they like. For more details of the batik pattern creation, refer to the project collaborator's part.

Batik Patterns Customization

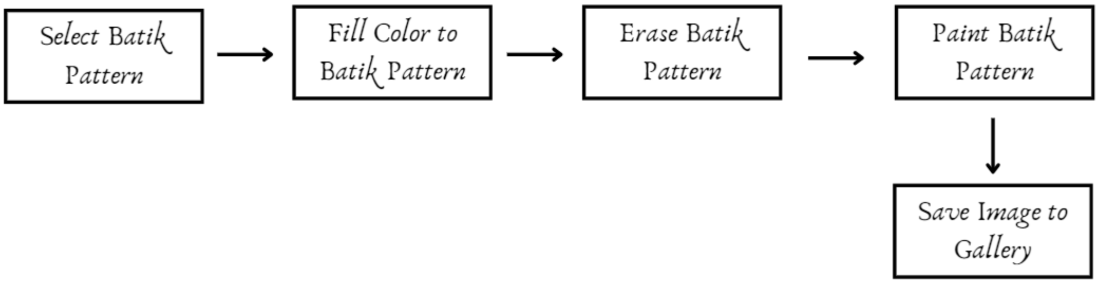


Figure 4.2.4 Component Specifications Diagram of Batik Patterns Customization

For the batik pattern customization, users have to choose the batik pattern that they want to customize either from the internal app gallery or the phone gallery. After the batik pattern is selected, it will upload to the interface, showing the edge of the batik for further filling color, painting, or erase purposes. Then, the user may start to customize the batik pattern based on their preference, and the final result may be saved into the gallery afterwards. For more algorithm information, refer to the project's collaborator part.

Gallery

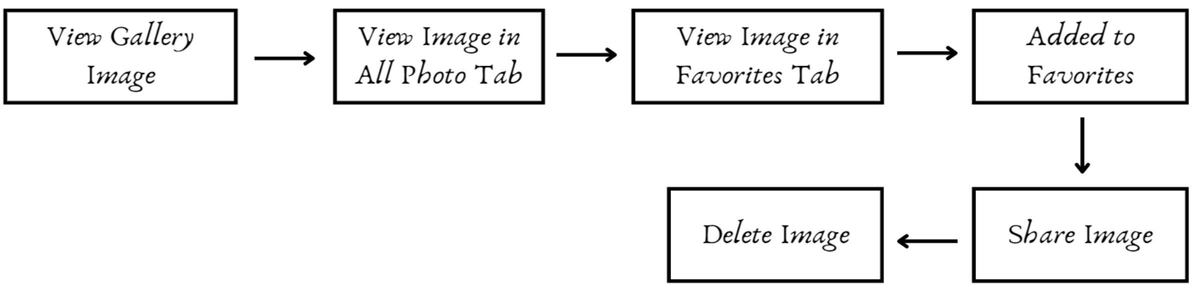


Figure 4.2.5 Component Specifications Diagram of Gallery

This figure shows where a user views an image from their gallery. After selecting a gallery image, the user navigates to the "All Photo" tab to view the image. This tab likely contains all the images stored in the gallery. The user can then move to the "Favorites" tab, where images that are marked as favorites are displayed. This tab is specifically for images that the user has marked as important or favorite. When viewing an image in the “All Photo” or "Favorites" tab, user has the option to share the image with others, possibly via social media or other sharing platforms such as WhatsApp, Messenger and others. Alternatively, after viewing the image in either one of the tab, user has also the option to delete the image from the gallery.

Tutorials

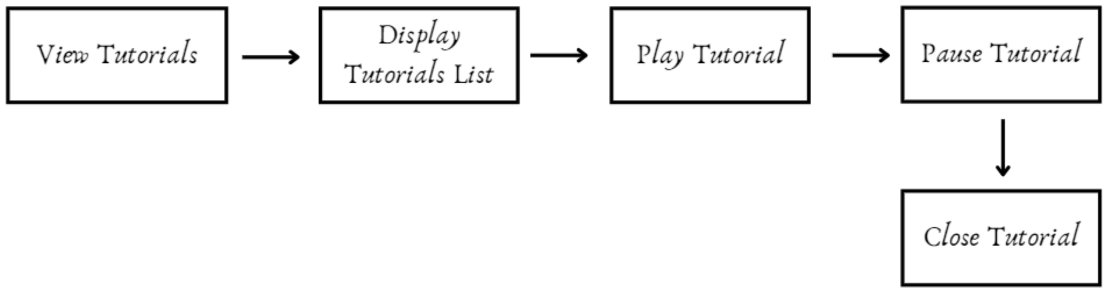


Figure 4.2.6 Component Specifications Diagram of Tutorials

The flowchart represents the process of interacting with tutorials in an application. The flow begins when a user selects the option to View Tutorials through a hint button, which prompts the application to Display the Tutorials List. From this list, the user can choose a specific tutorial to begin playing by clicking the Play Tutorial action. While the tutorial is playing, the user has the ability to Pause the Tutorial if they need to stop temporarily. Once paused, the user can choose to resume the tutorial afterwards while also may decide to Close the Tutorial anytime, which ends the session and exits the tutorial. This flow represents a typical user interaction when consuming tutorial content, which provides a guideline on how to play through with the application, providing the ability to start, pause, and close a tutorial as needed.

Chapter 5 System Implementation

5.1 Hardware Setup

The hardware being used in this project is a laptop, mobile device and a ubuntu server. The laptop and ubuntu server is being used to run the GAN model, which could generate the try-on result of the user, while the mobile device is being used to insert the input of the person's and cloth's image, and the user will finally get the generated virtual try-on result image shown on the phone.

Description	**Specifications**
Model	MSI Modern 14 B11MO
Processor	11^{th} Intel Core i5-115G7
Operating System	Windows 11 64-bit
Graphic	Intel® Iris® Xe Graphic
Memory	8192 MB RAM
Storage	476 GB

Table 5.1.1 Specifications of laptop

Description	**Specifications**
Model	MSI
Processor	Intel Core i9-13900H
Operating System	Window 11
Graphic	NVIDIA® GeForce RTX™ 4070
Memory	16 GB RAM
Storage	1TB

Table 5.1.2 Specifications of Ubuntu Server on Another Laptop

5.2 Software Setup

1. **Android Studio**

The software being used in this project is Android Studio, which acts as a platform to design the user interface and perform interaction between users and backend server.

2.Ubuntu

Ubuntu is used as the server environment to handle backend processing and image-related tasks. It supports essential tools and libraries needed for running model that require GPU environment and communication with the Android client.

5.3 Settings and Configuration

5.3.1 Android Studio Settings and Configuration

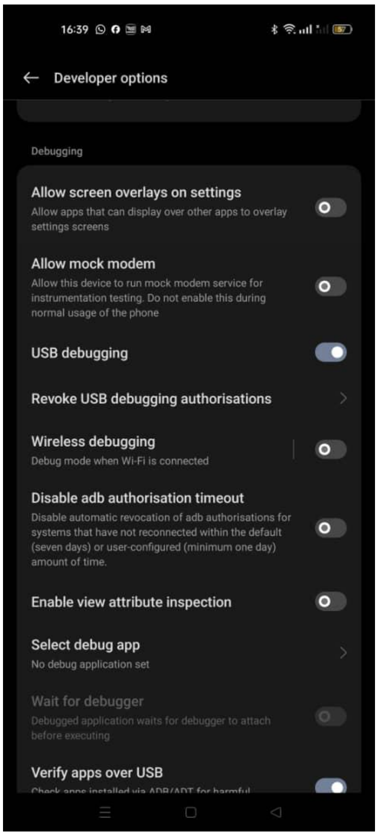


Figure 5.3.1 Developer Options – Wireless Debugging

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE"
    tools:ignore="ScopedStorage" />
<uses-permission android:name="android.permission.INTERNET" />
```

Figure 5.3.2 User Permissions in AndroidManifest.xml

```
dependencies {
    implementation project(':sdk4') // Reference to the OpenCV SDK module
    implementation 'com.squareup.retrofit2:retrofit:2.9.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
    implementation 'com.github.bumptech.glide:glide:4.15.1'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.15.1'
    implementation 'com.google.android.material:material:1.9.0'
    implementation 'com.squareup.retrofit2:converter-scalars:2.9.0'
    implementation 'com.github.yukuku:ambilwarna:2.0.1'
    implementation 'androidx.cardview:cardview:1.0.0'
    implementation 'com.github.bumptech.glide:glide:4.12.0'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0'

}
```

Figure 5.3.3 Dependencies in build.gradle

The above figures show the settings and configurations in Android Studio. For debugging purposes, the testing phone is set to USB debugging mode under the phone's developer option, as debugging in a real-time phone can provide a more realistic point of view instead of using the emulator provided by Android Studio. Besides, the manifest file and gradle file have to be filled in with all the requirements shown above in the figure in order to run the application successfully.

5.3.2 Flask Settings and Configuration

Figure 5.3.4 Visual Studio Code – Install Python

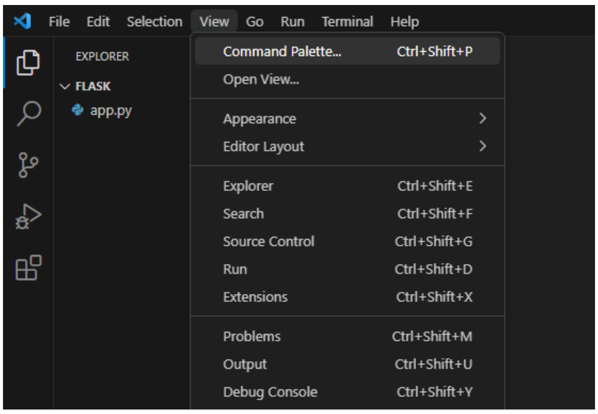


Figure 5.3.5 Visual Studio Code – Create app.py

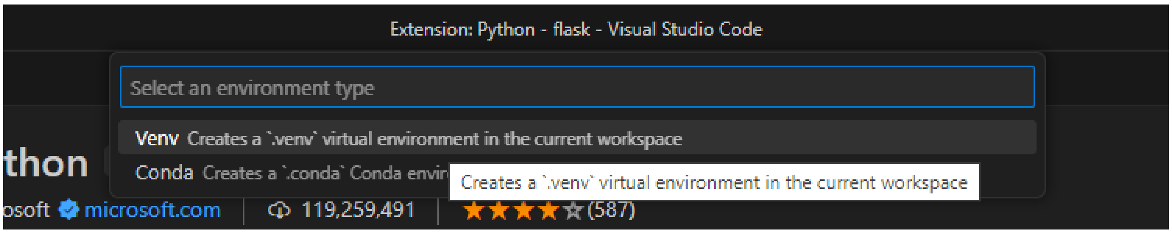


Figure 5.3.6 Visual Studio Code – Create Virtual Environment

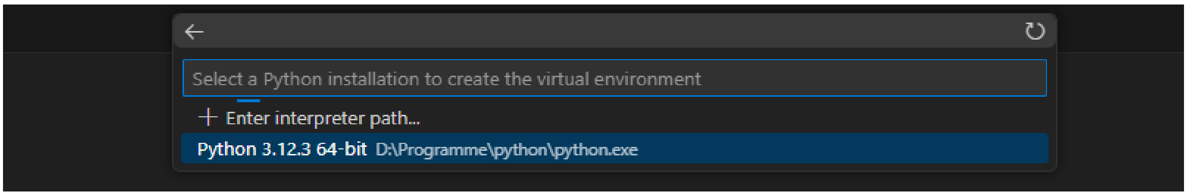


Figure 5.3.7 Visual Studio Code – Choose Python as Interpreter

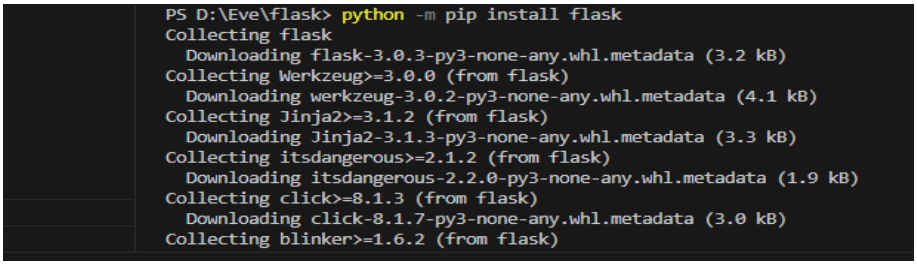


Figure 5.3.8 Visual Studio Code – Install Flask

The above figures show a series of steps on how to set up the Visual Studio Code as Flask and act as an intermediate platform to connect between Android Studio and Ubuntu Server. It has to install Python dependencies when first entering Visual Studio Code. Then, search the command palette and try to find to create an environment. After that, install the appropriate interpreter and also the Flask (at the terminal), which has done the set-up of Flask in Visual Studio Code.

5.4 System Operations

5.4.1 Connection Between Android Studio, Flask and Ubuntu

```java
// Create Retrofit client
Retrofit retrofit = new Retrofit.Builder()
        .baseUrl("http://192.168.1.32:5000/")
        .addConverterFactory(ScalarsConverterFactory.create())
        .build();

VirtualTryService apiService = retrofit.create(VirtualTryService.class);
```

Figure 5.4.1 Android Studio – Retrofit Client

```python
@app.route('/trigger-segment-cloth/', methods=['POST'])
def trigger_segment_cloth(base64_image, value_name):
    """Send the resized image and value_name to the Ubuntu server in Base64 format."""
    ubuntu_server_url = 'http://172.17.108.226:5001/segment-cloth/'
    try:
        print("Triggering segmentation cloth process.")
        data = {'image': base64_image, 'value_name': value_name}

        response = requests.post(ubuntu_server_url, json=data)
        print(f"Segmentation request sent. Status Code: {response.status_code}")

        if response.status_code == 200:
            print("Segmentation triggered successfully on Ubuntu.")
            return jsonify({"Status": "Triggered successfully on Ubuntu"}), 200
        else:
            print(f"Failed to trigger segmentation. Detail: {response.text}")
            return jsonify({"Status": "Failed to trigger on Ubuntu", "Detail": response.text}), response.status_code
    except Exception as e:
        print(f"Error in trigger_segment_cloth: {e}")
        return jsonify({"Status": "Error", "Detail": str(e)}), 500
```

Figure 5.4.2 Flask – trigger to Ubuntu Server

```python
@app.route('/image-latest/', methods=['GET'])
def getting_latest_image():
    folder = "//wsl.localhost/Ubuntu/home/solomon/vton/VTonHD/VITON-HD-main/results/Testing"
    try:
        print("Fetching latest image.")
        all_images = sorted(
            [f for f in os.listdir(folder) if os.path.isfile(os.path.join(folder, f))],
            key=lambda x: os.path.getmtime(os.path.join(folder, x)),
            reverse=True
        )

        if not all_images:
            print("No images found in the folder.")
            return jsonify({"error": "No images found in the folder"}), 404

        latest_image = all_images[0]
        image_path = os.path.join(folder, latest_image)
        print(f"Latest image found: {latest_image}")

        with open(image_path, "rb") as image_file:
            encoded_string = base64.b64encode(image_file.read()).decode('utf-8')
            print("encoded",encoded_string)

        return jsonify({
            "imageFile": encoded_string,
            "imageName": latest_image
        }), 200

    except Exception as e:
        print(f"Error in getting_latest_image: {e}")
        return jsonify({"error": str(e)}), 500
```

Figure 5.4.3 Ubuntu – Waiting for Trigger Result

```java
1 usage
private void getLatestImage() {
    // Create Retrofit client for getting the latest image
    Retrofit retrofit = new Retrofit.Builder()
            .baseUrl("http://192.168.1.32:5000/")
            .addConverterFactory(GsonConverterFactory.create()) // Use Gson to handle JSON
            .build();

    VirtualTryService apiService = retrofit.create(VirtualTryService.class);
```

Figure 5.4.4 Android Studio – Trigger to Ubuntu Receive Result Image

The above figures show the main code of how to send a request to Flask, then Flask passes it to the Ubuntu, and finally, from the Android Studio, retrieve the back result from Ubuntu. At the Android Studio, it will first call the Flask machine by calling the IP address of its own machine. In the Flask, it will trigger to the Ubuntu side by calling the IP address of Ubuntu if needed to start running programs on Ubuntu. After that, Ubuntu will process function and store the result image in a file location. Android Studio will then trigger the IP address of Ubuntu, which can lead to retrieving the image that was saved in the file.

5.4.2 Application Interface

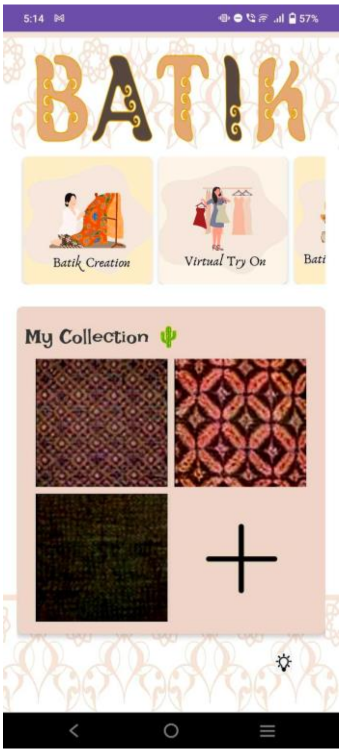


Figure 5.4.5 Main Menu of Batik Creator Application

Figure 5.4.5 shows the main menu of batik creator applications. The main menu shows the main function, where it has applied a scroll effect for the functions such as batik creation, virtual try-on, batik customization, and batik cloth transfer, which enable some of the functions to be temporarily hidden where the user needs to scroll left or right to see. The user is able to scroll left or right and click into the function page that they want to access. Besides, there is also a gallery function that occupies a large area on the screen and initially will display few pictures. However, if the user needs to view more, they need to click into it. Besides, there is also a hint button at the right corner of the main menu where this is the tutorials function that provides the user with a guideline video on how to use this application and how each function intends to perform.

Figure 5.4.6 Batik Creation Page of Batik Creator Application

The above figure shows the interface of the batik creation page, where there will be a generate button that will generate out an amount of batik patterns and display them inside the "Patterns" border. When the user clicks on either one of the batiks, it will display at upper side that has a bigger view of the batik pattern. Besides, the right side of the page provides several functions for the user to perform, such as adjust complexity, saturation, contrast, and brightness of the batik image, which aim to let the user play around with the batik image. There is also a save button that lets the user store the batik image that they like in the gallery, while there is also a button that can lead to another page, which is the Virtual Try-On page for playing around on virtual try-on.

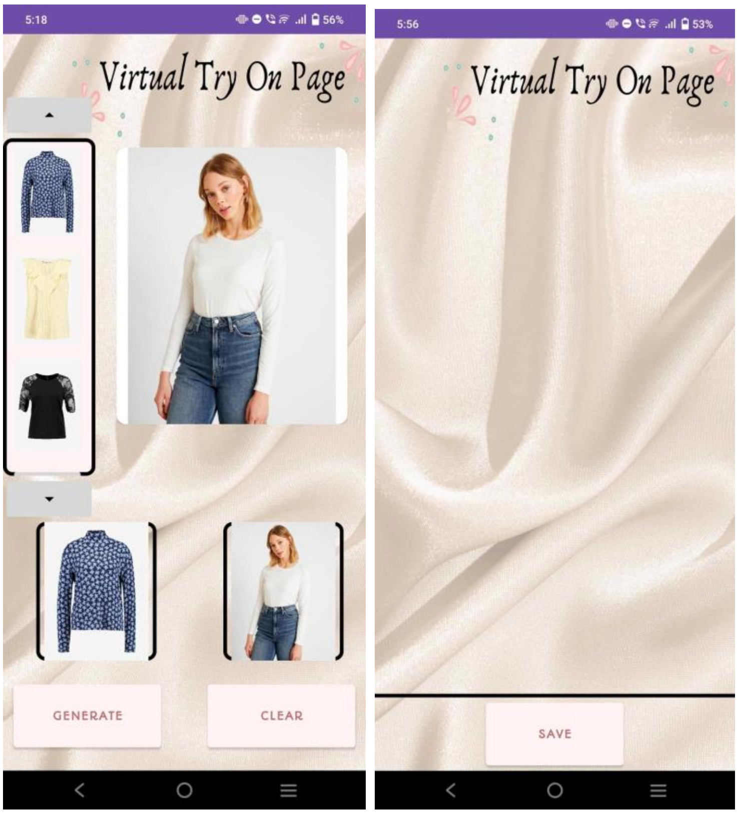


Figure 5.4.7 Virtual Try-On Page of Batik Creator Application

Figures above show the virtual try-on page, where at the left side there is a container consisting of a few cloth images for the user's choosing. However, there is also an add button at the bottom of container that is able to let the user choose cloth from either the app gallery or the phone gallery. User may scroll up and down the clothes through button, while also for the person model, user may scroll left and right to view more person models. For the selected cloth and person, the model will be displayed at the respective border. There is a generate button and a clear which allow the user to generate the try on result, and the result image will be displayed into the border above the save button, while also the clear button is in charge of clearing out all the images in the border displayed view. There is also a save button at the bottom of the page, which aims to let the user save the result into either an app gallery or a phone gallery based on their preferences. In this page, it applies the scroll view, which enables the user to scroll up and down through the whole screen.

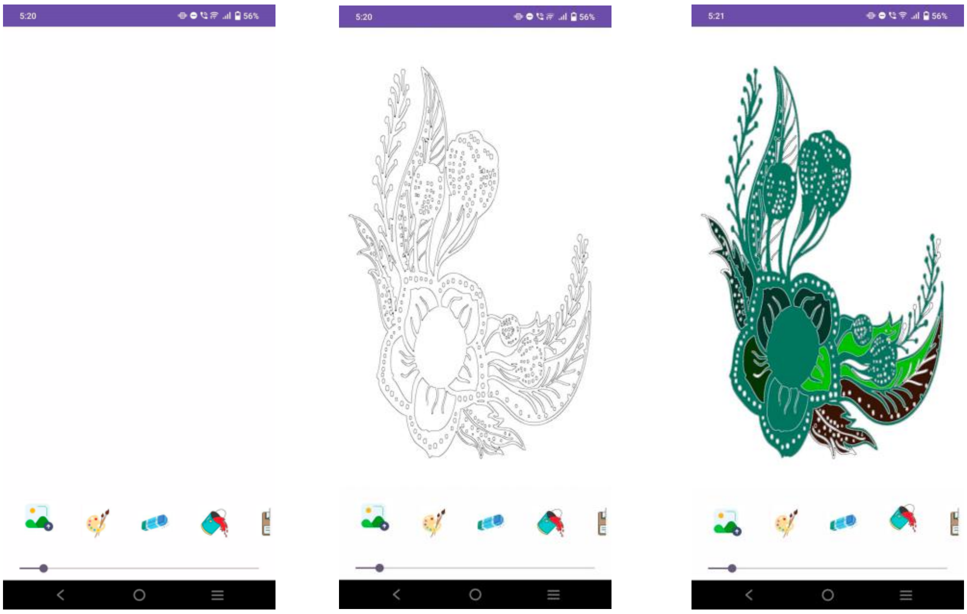


Figure 5.4.8 Batik Customization Page of Batik Creator Application

In the batik customization page, it has a row of buttons at the bottom of the screen that contain an upload image button, a fill color button, an erase button, a paint button, and a save button. Each of the buttons carries out different functions. However, users first have to upload an image so that they can just proceed to the following steps, such as filling or painting an image, and others. This page aims to provide a fun experience to users where they can edit the batik pattern themselves based on their preferences.

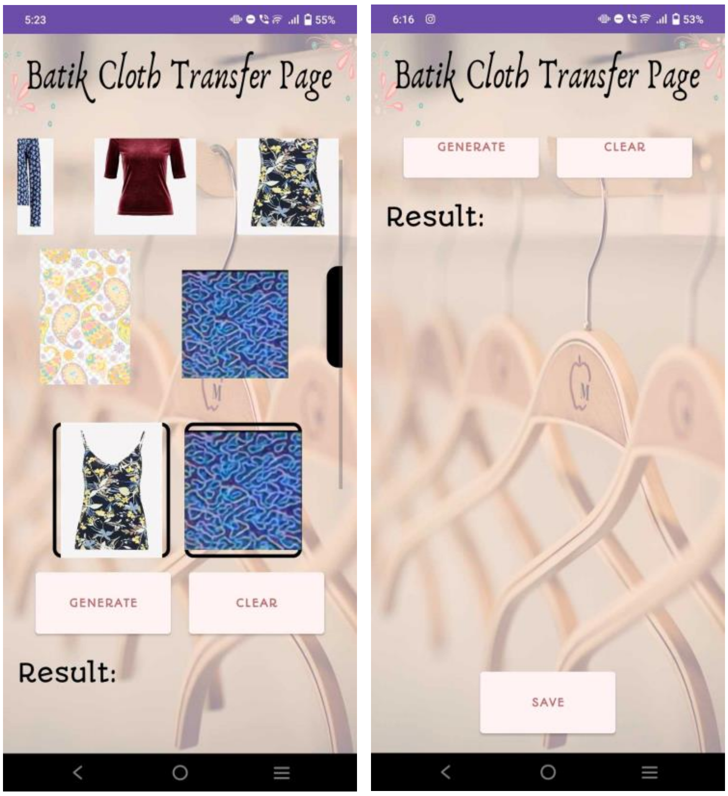


Figure 5.4.9 Batik Cloth Transfer Page of Batik Creator Application

For the batik cloth transfer page, there are two rows at the top of the screen where the first row displays cloth while the second row is displayed batik. Two rows can be scrolled left or right to view more images, while also for the batik row, it allows the user to add more batik from either the app gallery or the phone gallery through the add more button at the end of the row. The selected cloth or batik pattern will be displayed into a border, and there is also a generate button and clear button that are in charge of generating the batik cloth transfer result and clearing all images in the border. At the bottom of the screen there is also a save button to allow the user save the result to the gallery, while also a border that will display the final result of batik transfer to cloth. In this page, the scroll view also applies through the whole screen so that the user can scroll around the whole screen.

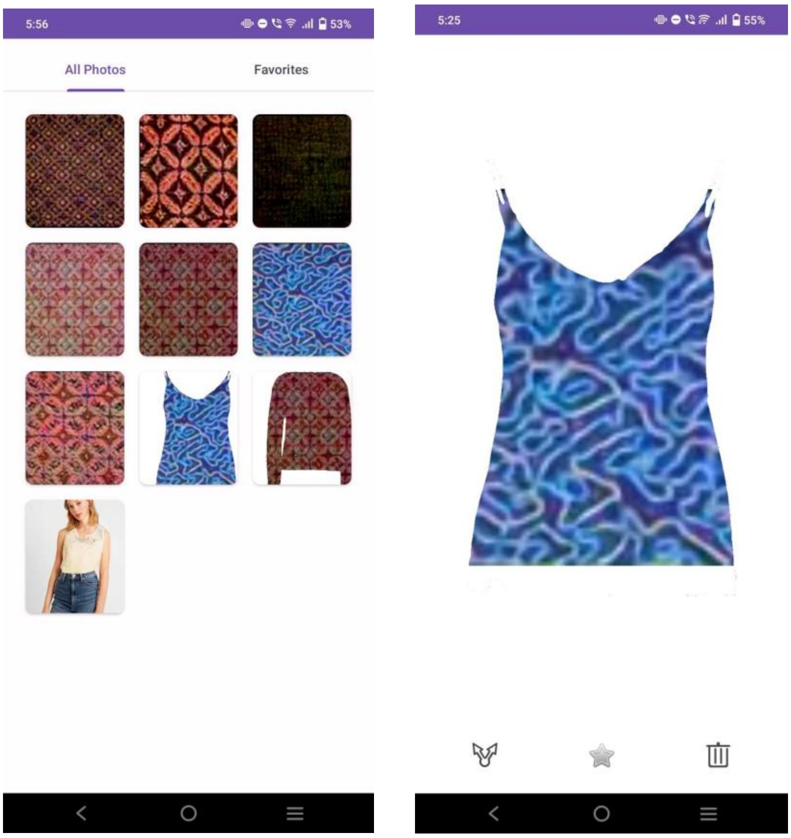


Figure 5.4.10 Gallery Page and Image Details of Batik Creator Application

In the gallery page, it has two tabs, one of which is the “All Photo” tab, which displays all the images that are being saved by users, while the “Favorites” tab is in charge of displaying image that are being added to “Favorites.”. The user may click on the image to see the details, and inside the details of the image there will be a row of buttons at the bottom, which are the share button, add to favorites button, and delete button, each of which is in charge of different functions.

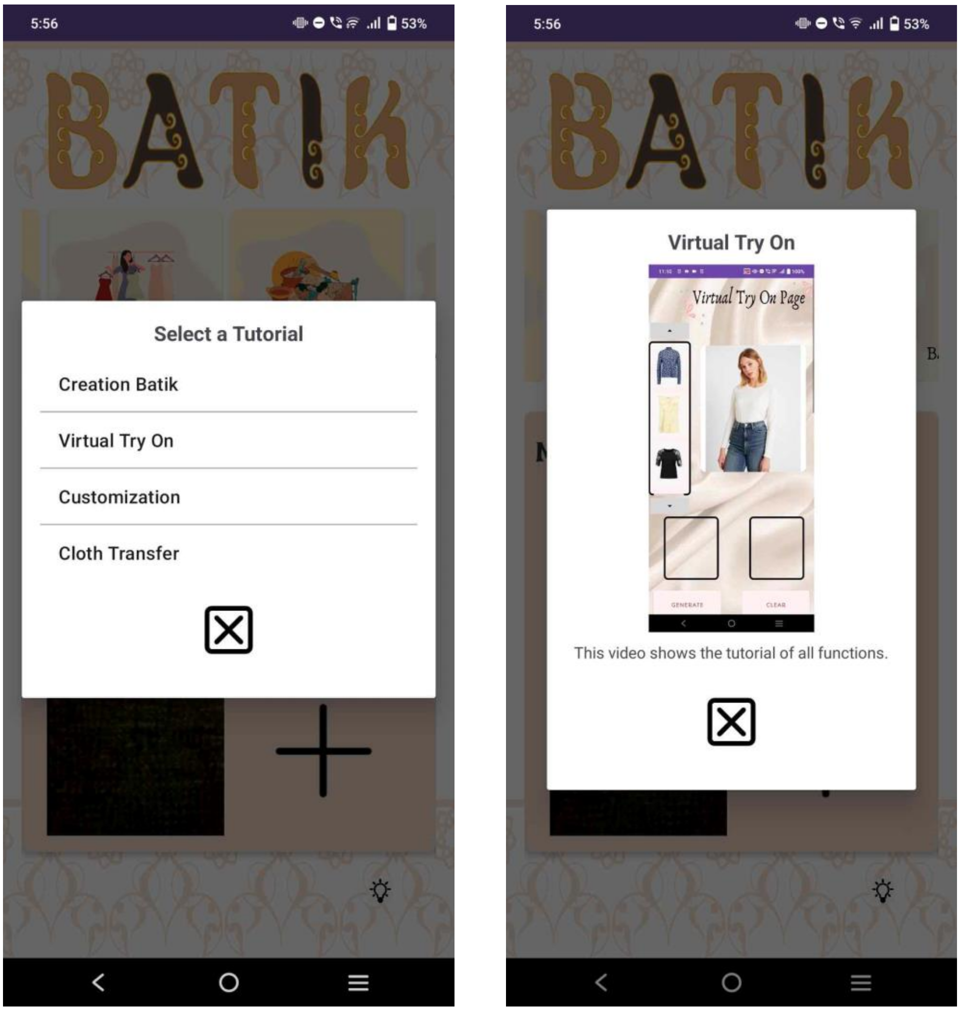


Figure 5.4.11 Tutorials Page of Batik Creator Application

For the tutorials page, when the user initially clicks on the hint button, which is displayed at the right corner of the main menu, it will prompt out a dialogue that will allow the user to choose whether they want to view the tutorials of which function of the application. After selecting, it will prompt to display video to the user, and the user may close the dialogue anytime.

5.5 Implementation and Challenges

The implementation issues and challenges facing the project are the automation real-time connection problem and the fact that we cannot get 100% accuracy of the virtual try-on result image. As the GAN model is a high computational power model that needed a GPU to run it, it was hosted on the cloud. The VITON-HD GAN model requires all input images to have a white background for accurate design overlay. This poses significant challenges: finding suitable person models is difficult due to the need for images with white backgrounds and precise keypoint annotations. Similarly, clothing images must also have a white background to ensure accurate application of batik designs. These constraints make sourcing and preparing appropriate images a complex task.

Figure 5.5.1 Unsuccessful Case of Virtual Try On

Chapter 6 System Evaluation and Discussion

6.1 System Testing and Performance Metrics

System testing and performance evaluation for the **Batik Creator Application** involve verifying the application operates as expected according to its specified requirements. The goal is to ensure that each component within the system functions seamlessly to meet user expectations. Since the application features multiple key functions, such as batik creation, batik cloth transfer, batik customization, virtual try-on, tutorials and a gallery, suitable system testing is crucial to identify potential issues. The interactions between these components can lead to defects or performance bottlenecks, so it is vital to thoroughly test each feature. At the conclusion of the testing phase, the results will be analyzed, and any issues found will be reported to the development team for prompt resolution, ensuring that the application is stable and ready for distribution to users.

To maintain the quality and reliability of the Batik Creator Application, various types of testing will be performed, with a focus on **unit testing**. Unit testing will break down the application into individual components, such as the batik creation tool or the gallery feature, and a checklist will be created for each. The checklist will contain specific test cases and actions needed to confirm that each feature whether it's generating a new batik design, customizing cloth patterns, or enabling users to virtually try on the designs functions as intended. Conducting these tests will help ensure that each feature is responsive, efficient, and bug-free, ultimately providing users with a smooth and intuitive experience.

6.2 Testing Setup and Result

In this section, each of the page will be carried out unit testing to ensure that the functions are work as expected.

Virtual Try-On Page

No	Test Case	Expected Result	Pass/Fail
1	Input cloth and batik only	When user click the generate button there	Pass

		will be a result display on the result view	
2	Choose a cloth from app gallery	Add in new picture into the cloth container	Pass
3	Scroll around the container with touch event without click on button	Able to scroll up and down through touch motion	Fail
4	Click Clear Button	Able to clear all the images that are displayed in the view	Pass
5	Trigger generate button with calling own machine ip address	Able to trigger to flask and perform function	Pass

Table 6.2.1 Test Result for Virtual Try-On Page

Batik Cloth Transfer Page

No	**Test Case**	**Expected Result**	**Pass/Fail**
1	Input batik and cloth image	Selected image displayed into correspond view	Pass
2	Choose batik from phone gallery	Add in new selected picture to the batik row	Pass
3	Scroll up and down the whole screen	Can view all content of the page	Pass
4	Click Clear Button	All image displayed in the view being cleared	Pass

5	Save Result Image	Able to save image to both phone and app gallery	Pass

Table 6.2.2 Test Result for Batik Cloth Transfer Page

Batik Creation Page

No	Test Case	Expected Result	Pass/Fail
1	Click Generate Button	Random batik pattern will be displayed to the screen	Pass
2	Adjust the complexity of batik pattern	Batik pattern able to transform from simple to complex	Pass
3	Adjust the brightness of batik	Batik can be more brighter or darkness	Pass
4	Click to go the Virtual Try On Page	Redirect to virtual try on page	Pass
5	Save favourite batik pattern into gallery	Image being saved to internal app gallery and phone gallery	Pass

Table 6.2.3 Test Result for Batik Creation Page

Batik Customization Page

No	Test Case	Expected Result	Pass/Fail
1	Upload Image through Upload Button	Display selected batik image into	Pass

		view with only showing the edge of batik pattern	
2	Paint the batik	Batik can be paint with colour one time with a large area	Pass
3	Erase the unwanted part on the batik	Unwanted part on the batik can be erased	Pass
4	Fill the batik with intended colow	Batik can be fill with any colour	Pass
5	Save the final batik result image into gallery	Result image can be saved to both phone gallery and app gallery	Pass

Table 6.2.4 Test Result for Batik Customization Page

Gallery

No	Test Case	Expected Result	Pass/Fail
1	View the image in "All Photo" tab	Image that saved by users displayed in the grid form	Pass
2	View the image being added as favourite	At the "Favorites" tab can display all favourites images	Pass
3	Share image	Able to share image to other media or platform	Pass

4	Add image as favourite	Image being added to "Favorites" section	Pass
5	Delete image	Deleted image will not be appear at the gallery	Pass

Table 6.2.5 Test Result for Gallery Page

Tutorials

No	Test Case	Expected Result	Pass/Fail
1	Click on the hint button on the main page	Able to show a tutorials dialog	Pass
2	Choose one of the tutorials want to view from the dialog list	Each tutorial able to lead to respective video	Pass
3	Play the tutorial video	Tutorial video will be automatically played when being clicked in	Pass
4	Pause the tutorial video	Tutorial video can be stopped at anytime	Pass
5	Close the dialog	Tutorials dialog can be closed	Pass

Table 6.2.6 Test Result for Tutorials Page

6.3 Project Challenges

1. Lack of resources

Due to the limitations of resource constraints which needed for high computational power of training on GAN model, this project is using the pre-trained GAN model. Although there is paper perform comparison of proven VITON-HD model has better performance compared to other model such as CP-VTON and ACGPN (shown at literature review). However, the VITON-HD model sometimes still generate the result that are not satisfied.

2. Response Time of Server

Another challenge in the Batik Creator Application is the server response time, especially when processing complex tasks such as virtual try-on. Since these features require sending images to the server for processing, the time it takes for the server to respond can significantly affect the user experience as the server has limited computational power.

6.4 Objectives Evaluation

The Batik Creator Application consists of six major tasks, each aimed to improve the user experience and provide a streamlined procedure for producing and managing batik patterns. When reviewing the application's objectives, the attention should be on the usability, efficiency, and quality of the features available to users. The first feature, batik generation, allows users to make batik patterns at random, with choices for varying the intricacy of the designs. It should be flexible in producing aesthetically beautiful patterns, and users should be able to refine their designs using filtering features. Furthermore, the option to save produced batik patterns is required, allowing users to conveniently store and revisit their ideas.

Users may transfer batik patterns onto a selected textile by using the batik cloth transfer function. This feature's accuracy and smoothness in applying the batik pattern to various fabrics should be evaluated, and it should also be checked that users are able to save the transferred picture in its final form. Evaluating the system should focus on how fast it can process the photos while preserving the design's visual integrity during the transfer.
The goal of batik customization is to allow customers the freedom to submit their own patterns and do unique edits, including sketching or modifying the designs. The success of this feature depends on how simple it is to use and how many customization options it gives

users for their projects. Additionally, the program needs to enable instantaneous response on user inputs, fostering creativity and intuitiveness in the customization process.

For the tutorials function, the lessons' thoroughness and clarity will be assessed to make sure users can utilize the app's capabilities, navigate it with ease, and solve any problems they may run into. Tutorials are provided in simple and understand with clear instructions.
Using the virtual try-on feature, customers may pick a cloth and a model to see how the chosen batik will seem on them. The evaluation would take into account the virtual try-on experience's authenticity and realism as well as how simple it is for consumers to save the finished photos. The goal of this feature should be to provide users an accurate idea of how the cloth would look on various models so they can make well-informed design decisions.
Lastly, the gallery acts as a storage facility for all of the user-generated photographs. Users should find it easy to share, organize, and remove the image. The gallery feature's usability is that if users can easily browse it and whether the sharing option works well with other platforms to make it easy for users to share their creations with others.

In summary, the Batik Creator Application's success depends on how well it achieves these objectives across its six functions, focusing on user experience**,** design quality, and system performance**.**

Chapter 7 Conclusion

7.1 Conclusions

In summary, the interaction between fashion and technology has transformed how consumers interact with clothing and accessories, especially when they purchase online. With the use of cutting-edge generative-AI techniques like GANs, virtual try-on technology can revolutionize the way the fashion industry presents and sells its items online. The compelling potential of GAN-based virtual try-on systems to close the gap between digital and physical retail settings, improve user experiences, and provide clothes sellers with cutting-edge tools for product showing is what drives this research. The project intends to provide an easy interface that integrates GAN models and allows users to virtually try on clothing items and see ideas in motion by emphasizing usability, simplicity, and creativity. Batik Creator Application bridges the gap between traditional shopping and virtual try on, providing a platform in which users can explore and engage with virtual try-on and batik designs in a way that reflects real-world processes. Besides, virtual try on add value by enabling users to visualize the look of their batik designs on the model, providing a realistic representation of the final product. From allowing users to randomly generate and customize batik creation features for complex patterns to seamlessly apply these designs to various fabric types of batik cloth transfer features, the app ensures flexibility and precision in the design process.

In addition, the library features improve the user experience by providing simple storage, organization, and sharing options for created images. Overall, Batik Creator Application has successfully performed virtual try-on and batik creation. With continuous optimization of performance and responsiveness, the application has strong potential for both casual users and professional designers.

7.2 Recommendations

To elevate the quality of the Batik Creator Application, enhancing both the user interface and the accuracy of the Virtual Try-On feature would be pivotal. Investing in a more aesthetically pleasing and intuitive UI design can significantly improve user experience by making navigation and customization processes more enjoyable and seamless. Additionally, refining the accuracy of the Virtual Try-On results will provide users with a more realistic and reliable visualization of their designs. By ensuring that the virtual representations closely match the actual appearance of the batik creations, users can make more informed decisions and feel more confident in their customizations. These improvements will not only enhance user satisfaction but also strengthen the app's reputation as a cutting-edge tool for batik design.

REFERENCES

[1] L. London, “Virtual Try-On is more than a pandemic trend and these brands are reaping the rewards,” Forbes, https://www.forbes.com/sites/lelalondon/2021/05/20/virtual-try-on-is-more-than-a-pandemic-trendand-these-brands-are-reaping-the-rewards/?sh=1105c3876c82 (accessed April. 19, 2024).

[2] L. Husband, “H&M launches virtual fitting room experience in Germany,” Just Style, https://www.just-style.com/news/hm-virtual-fitting-room-germany/?cf-view (accessed April. 19, 2024).

[3] H. Ghodhbani, M. Neji, I. Razzak, and A. M. Alimi, “You can try without visiting: A comprehensive survey on virtually try-on outfits - multimedia tools and applications,” SpringerLink, https://link.springer.com/article/10.1007/s11042-022-12802-6 (accessed April. 19, 2024).

[4] Ghodhbani, H. *et al.* (2022a) *You can try without visiting: A comprehensive survey on virtually try-on outfits - multimedia tools and applications*, *SpringerLink*. Available at: https://link.springer.com/article/10.1007/s11042-022-12802-6 (Accessed: 10 September 2024).

[5] SJSU ScholarWorks, Virtual Try-On With Gener y-On With Generative Adversarial Networks: A ersarial Networks: A Taxonomical Sur onomical Survey, https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=2612&context=faculty_rsca (accessed Sep. 11, 2023).

[6] N. Jetchev and U. Bergmann, “The conditional analogy gan: Swapping fashion articles on people images,” *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017. doi:10.1109/iccvw.2017.269

[7] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, “Viton: An image-based Virtual Try-On Network,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. doi:10.1109/cvpr.2018.00787

[8] S. Zhu, S. Fidler, R. Urtasun, D. Lin, and C. C. Loy, “Be your own prada: Fashion synthesis with structural coherence,” *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. doi:10.1109/iccv.2017.186

[9] S. He, Y.-Z. Song, and T. Xiang, “Style-based global appearance flow for virtual try-on,” *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. doi:10.1109/cvpr52688.2022.00346

[10] S. Choi, S. Park, M. Lee, and J. Choo, “Viton-HD: High-resolution virtual try-on via misalignment-aware normalization,” *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021. doi:10.1109/cvpr46437.2021.01391

[11] *"HeyBeauty - AI Virtual Try-On App," Heybeauty.ai, 2024. https://heybeauty.ai/ (accessed Sep. 10, 2024).*

[12] *"Style.me | A New Way to Experience Digital Fashion," Style.me. https://style.me/*

[13] "PICTOFiT Leading Virtual Try-On Platform - Reactive Reality," Jul. 01, 2022. https://www.reactivereality.com/pictofit

[14] S. Choi, "shadow2496/VITON-HD," GitHub, Sep. 25, 2023. https://github.com/shadow2496/VITON-HD

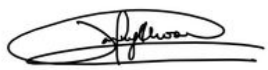
[15] wildoctopus, "GitHub - wildoctopus/huggingface-cloth-segmentation: Huggingface cloth segmentation using U2NET," GitHub, 2023. https://github.com/wildoctopus/huggingface-cloth-segmentation (accessed Sep. 10, 2024).

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.: 3**
Student Name & ID: Teo Zi Ning 2104495	
Supervisor: Dr Tan Hung Khoon	
Project Title: Virtual Try-On Creation using Generative-AI	

1. WORK DONE [Please write the details of the work done in the last fortnight.] Review existing systems.
2. WORK TO BE DONE Design application interface.
3. PROBLEMS ENCOUNTERED None
4. SELF EVALUATION OF THE PROGRESS Keep making progress once have free time.



Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.: 6**
Student Name & ID: Teo Zi Ning 2104495	
Supervisor: Dr Tan Hung Khoon	
Project Title: Virtual Try-On Creation using Generative-AI	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

User interface complete designed.

2. WORK TO BE DONE

Do batik transfer to cloth.

3. PROBLEMS ENCOUNTERED

When testing on cloth with different color background cannot work while also unable to try out different person model.

4. SELF EVALUATION OF THE PROGRESS

Keep making progress once have free time.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.: 9**
Student Name & ID: Teo Zi Ning 2104495	
Supervisor: Dr Tan Hung Khoon	
Project Title: Virtual Try-On Creation using Generative-AI	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Successful have code on how batik can transfer to cloth.

2. WORK TO BE DONE

Integrate my part with collaborator part into one application.

3. PROBLEMS ENCOUNTERED

None.

4. SELF EVALUATION OF THE PROGRESS

Keep making progress once have free time.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.: 11**
Student Name & ID: Teo Zi Ning 2104495	
Supervisor: Dr Tan Hung Khoon	
Project Title: Virtual Try-On Creation using Generative-AI	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Done almost system code.

2. WORK TO BE DONE

Left few parts of code to be debugged and need to start to do report.

3. PROBLEMS ENCOUNTERED

Calling API.

4. SELF EVALUATION OF THE PROGRESS

Collaborate with collaborator on combining integration part.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.: 13**
Student Name & ID: Teo Zi Ning 2104495	
Supervisor: Dr Tan Hung Khoon	
Project Title: Virtual Try-On Creation using Generative-AI	

1. WORK DONE [Please write the details of the work done in the last fortnight.] Done Report.
2. WORK TO BE DONE None.
3. PROBLEMS ENCOUNTERED None.
4. SELF EVALUATION OF THE PROGRESS None.

Supervisor's signature

Student's signature

POSTER

VIRTUAL TRY-ON CREATION USING GENERATIVE-AI
01 Introduction
Fashion and technology have merged, transforming how consumers engage with apparel. Online shopping faces challenges due to the inability to try clothes, leading to significant return issues for businesses. Virtual try-on technology powered by unique AI solutions allows users to digitally try on garments, bridging the gap between physical and digital shopping. Brands like H&M use augmented reality to enhance this experience. Virtual try-on relies on techniques like garment transfer and human pose estimation, facilitated by advancements in generative AI such as GANs. This project explores the VITON-HD GAN model to illustrate how virtual try-on systems revolutionize digital fashion experiences, enabling lifelike interactions with clothing online.
02 Objectives
Random Batik Pattern Generation: The app will use generative algorithms to create and explore diverse batik patterns interactively.
Batik Cloth Transfer and Virtual Try-On: It will allow users to apply batik designs to cloth and visualize them on models, enhancing virtual try-on realism.
VITON-HD Model Integration: The app will use the VITON-HD GAN model for a highly realistic virtual try-on, improving user engagement and aiding fashion businesses.
03 Methodology
Develop a virtual try-on application enabling users to visualize clothing on virtual models through an intuitive interface.
Utilize Firebase for storing user inputs and Google Colab as a cloud function to retrieve inputs and generate virtual try-on images using the VITON-HD GAN model.
Implement texture transfer features allowing users to transform cloth textures and integrate a recommendation system to suggest nearby clothing shops based on location. This project aims to enhance user engagement and interaction with virtual fashion through innovative technology integration.
04 Contributions
Pre-processing involves resizing images to 768x1024 and generating cloth masks, ensuring compatibility with the GAN model for accurate virtual try-on results. This study applies GAN technology to enhance virtual try-on systems, improving user interaction and realism in clothing simulations.
05 Conclusion
Fashion and technology have transformed online shopping experiences, with generative-AI techniques like GANs offering virtual try-on systems that bridge the gap between digital and physical retail. This research focuses on integrating GAN models into a user-friendly interface for virtual clothing trials, emphasizing simplicity and creativity. Challenges like real-time connectivity and image quality remain to be addressed in future project phases.
Project Developer: Teo Zi Ning
Project Supervisor: Ts Dr Tan Hung Khoon

PLAGIARISM CHECK RESULT

check turnitin.docx

ORIGINALITY REPORT

6%	4%	2%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	3%
2	Submitted to University of the Arts, London Student Paper	1%
3	Submitted to Universiti Tunku Abdul Rahman Student Paper	1%
4	Sihan Huang, Baicun Wang, Xingyu Li, Pai Zheng, Dimitris Mourtzis, Lihui Wang. "Industry 5.0 and Society 5.0—Comparison, complementation and co-evolution", Journal of Manufacturing Systems, 2022 Publication	1%
5	Zubair Ahmad Parrey, Sajad Hussain Shah, Firoz Mohammad, Manzer H. Siddiqui, Saud Alamri, Hazem M. Kalaji. "Exogenous epibrassinolide application improves essential oil biosynthesis and trichome development in peppermint via modulating growth and physicochemical processes", Research Square Platform LLC, 2023 Publication	<1%

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Teo Zi Ning
ID Number(s)	2104495
Programme / Course	Computer Science
Title of Final Year Project	Virtual Try-On Using Generative-AI

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 6 **%** **Similarity by source** Internet Sources: 4 % Publications: 2 % Student Papers: 3 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

	-
Signature of Supervisor	Signature of Co-Supervisor

Name: Tan Hung Khoon

Date: 11/9/2024

Name: -

Date: -

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	2104495
Student Name	Teo Zi Ning
Supervisor Name	Ts Dr Tan Hung Khoon

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)

Date: 11 September 2024